

UNIVERSITÉ PARIS-EST  
**ÉCOLE DOCTORALE MSTIC**

**T H È S E**

pour obtenir le titre de

Docteur de l'Université Paris-Est  
**spécialité : Informatique**

Présentée et soutenue par

**Ines AYADI**

**Optimisation des politiques de maintenance préventives  
dans un cadre de modélisation par modèles graphiques  
probabilistes**

Soutenue le 29/08/2013

préparée à l'Institut Français des Sciences et Technologies  
des Transports, de l'Aménagement et des Réseaux

**Jury :**

Mme. Anne Barros	Professeur, UTT	Rapporteur
M. Arnaud de La Fortelle	Professeur, Mines ParisTech	Rapporteur
M. Abderrahman El Mhamedi	Professeur, Paris 8	Examineur
M. Laurent Bouillaut	Chargé de Recherche, IFSTTAR	Examineur
M. Patrice Aknin	Directeur de Recherche, IFSTTAR	Co-directeur de thèse
M. Patrick Siarry	Professeur des Universités, UPEC	Co-directeur de thèse



# Table des matières

<b>Introduction Générale</b>	<b>vii</b>
<b>1 Modélisation de la dynamique de dégradation</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.2 Modèles à dégradation continue . . . . .	2
1.2.1 Le processus de Wiener . . . . .	3
1.2.2 Le processus Gamma . . . . .	7
1.2.3 Le processus de Poisson composé . . . . .	9
1.3 Modèles à dégradation discrète . . . . .	10
1.3.1 Processus de poisson . . . . .	11
1.3.2 Chaînes de Markov . . . . .	13
1.3.3 Processus Semi-markovien . . . . .	14
1.4 Modèles graphiques probabilistes temporels . . . . .	16
1.4.1 Réseaux bayésiens dynamiques . . . . .	17
1.4.2 Modèle graphique de durée . . . . .	26
1.5 Conclusion . . . . .	34
<b>2 Évaluation des stratégies de maintenance</b>	<b>35</b>
2.1 Introduction . . . . .	35
2.2 Enjeux de la fonction Maintenance . . . . .	36
2.3 Définition de la maintenance . . . . .	36
2.3.1 Fiabilité . . . . .	37
2.3.2 Maintenabilité . . . . .	38
2.3.3 Disponibilité . . . . .	38
2.3.4 Durabilité . . . . .	38
2.4 Théorie de la maintenance . . . . .	39
2.4.1 Maintenance corrective . . . . .	39
2.4.2 Maintenance préventive . . . . .	40
2.4.3 Maintenance mixte . . . . .	43
2.4.4 Maintenance opportuniste . . . . .	44
2.4.5 Maintenance imparfaite . . . . .	44
2.5 Modélisation de la maintenance . . . . .	45
2.5.1 Description du modèle VirMaLab . . . . .	46
2.5.2 Paramètres du modèle VirMaLab . . . . .	48
2.6 Évaluation des politiques de la maintenance . . . . .	48
2.6.1 Disponibilité du système . . . . .	49
2.6.2 Diagnostic . . . . .	49

2.6.3	Action de maintenance . . . . .	51
2.6.4	Critères d'évaluation d'une politique de maintenance . . . . .	51
2.7	Conclusion . . . . .	56
<b>3</b>	<b>Les métaheuristiques : état de l'art</b>	<b>59</b>
3.1	Introduction . . . . .	59
3.2	Les méthodes de résolution . . . . .	60
3.2.1	Méthodes exactes . . . . .	60
3.2.2	Méthodes approchées . . . . .	62
3.3	L'optimisation multiobjectif . . . . .	66
3.3.1	Formulation générale d'un problème d'optimisation multiob- jectif . . . . .	66
3.3.2	Multiplicité des solutions . . . . .	67
3.3.3	Dominance et front de Pareto . . . . .	67
3.3.4	Catégories d'algorithmes . . . . .	69
3.4	Transformation du POM en un problème mono-objectif . . . . .	70
3.4.1	Méthodes d'agrégation . . . . .	70
3.4.2	Méthode $\epsilon$ -contrainte . . . . .	70
3.4.3	Les algorithmes génétiques . . . . .	71
3.4.4	La recherche tabou . . . . .	75
3.4.5	Le recuit simulé . . . . .	75
3.4.6	Algorithmes basés sur l'intelligence collective : Les colonies de fourmis . . . . .	77
3.5	Approches non Pareto . . . . .	77
3.5.1	Algorithme VEGA . . . . .	77
3.5.2	Utilisation des genres . . . . .	78
3.6	Approches Pareto . . . . .	78
3.6.1	NSGA-II . . . . .	78
3.6.2	SPEA2 . . . . .	79
3.6.3	$\epsilon$ -MOEA . . . . .	80
3.6.4	MO-CMA-ES . . . . .	81
3.7	Conclusion . . . . .	81
<b>4</b>	<b>Optimisation de la maintenance préventive</b>	<b>83</b>
4.1	Introduction . . . . .	83
4.2	Généralités sur l'approche . . . . .	84
4.3	Codage de la solution candidate . . . . .	84
4.4	Initialisation des solutions . . . . .	86
4.5	Mesure de qualité de la solution candidate . . . . .	87
4.6	Choix de la taille de la population . . . . .	88

4.7	Algorithme génétique pour l'optimisation de la maintenance préventive périodique AG-MP(P)	88
4.7.1	Initialisation des solutions et limites de l'espace de recherche	88
4.7.2	Déroulement de l'algorithme	89
4.7.3	Sélection d'individus	90
4.7.4	Opérateurs de reproduction	91
4.7.5	Maintenir la diversité dans la population	92
4.8	Algorithme génétique pour l'optimisation de la maintenance préventive non périodique AG-MP(NP)	93
4.8.1	Déroulement de l'algorithme	93
4.8.2	Croisement	96
4.8.3	Mutation	99
4.8.4	Maintenir la diversité de la population	106
4.8.5	Condition d'arrêt	106
4.9	Recuit simulé modifié (RSM) pour l'optimisation de la maintenance préventive	109
4.9.1	Déroulement de l'algorithme	109
4.9.2	Paramètres	110
4.9.3	Résultats	113
4.10	L'algorithme ev-MMOGA pour l'optimisation de la maintenance préventive	114
4.10.1	Subdivision dynamique de l'espace des objectifs	115
4.10.2	Le concept de $\varepsilon$ -dominance	116
4.10.3	Déroulement de l'algorithme	117
4.10.4	Choix et adaptation d' $\varepsilon$	120
4.10.5	Opérateurs de reproduction adaptés	121
4.11	NSGA-II modifié pour l'optimisation de la maintenance préventive périodique	125
4.11.1	Description générale de l'algorithme	125
4.11.2	Distance de surpeuplement dynamique	126
4.12	Conclusion	127
<b>5</b>	<b>Optimisation de la maintenance préventive des rails</b>	<b>129</b>
5.1	Introduction	129
5.2	Contexte de l'étude	130
5.3	Modélisation de la dynamique de dégradation de la voie	132
5.3.1	Représentation de la voie	132
5.3.2	Modèle de dégradation du coupon	134
5.3.3	Politiques de maintenance modélisées	137
5.4	Optimisation mono-objectif de la MP des rails	140

5.4.1	Objectif de disponibilité . . . . .	141
5.4.2	Objectif budgétaire . . . . .	141
5.4.3	Paramètres de l'étude pour l'optimisation mono-objectif . . .	142
5.4.4	Algorithme génétique . . . . .	144
5.4.5	Recuit simulé modifié . . . . .	146
5.4.6	Discussion . . . . .	148
5.5	Optimisation multi-objectif : application à la MP des rails . . . . .	148
5.5.1	Fonctions objectifs . . . . .	148
5.5.2	NSGA-II modifié . . . . .	149
5.5.3	Ev-MMOGA . . . . .	150
5.5.4	Comparaisons des algorithmes . . . . .	151
5.6	conclusion . . . . .	154
<b>Conclusion et perspectives</b>		<b>157</b>
<b>Bibliographie</b>		<b>161</b>

# Introduction Générale

Depuis une dizaine d'années, les activités liées à la sûreté de fonctionnement ont pris une place prépondérante dans le domaine de l'industrie. Par ailleurs, afin d'assurer le maintien du système en fonction et sans incidents majeurs, l'intérêt de disposer de stratégies et de moyens de prévention demeure de plus en plus vital.

Dans ce cadre, de nombreuses études académiques et industrielles ont été menées dans le domaine de la fiabilité et de la sûreté de fonctionnement. Ces travaux se sont intéressés essentiellement à l'étude des systèmes et de leur dynamique de dégradation, en vue de décider des opérations de maintenance préventive les plus adéquates à entreprendre pour limiter les défaillances des systèmes.

Néanmoins, cette prise de décision n'est pas souvent évidente compte tenu du nombre important d'actions de maintenance, des contraintes budgétaires et logistiques et de l'horizon temporel de l'étude. Il devient alors indispensable de disposer d'outils et de méthodes permettant d'aider les décideurs pour une meilleure planification et optimisation des procédures de maintenance.

Les travaux de cette thèse partent par ailleurs du constat que dans plusieurs applications de l'industrie, le besoin de stratégies de maintenance assurant à la fois une sécurité optimale et une rentabilité maximale demeure de plus en plus croissant. En outre, ce besoin conduit à se référer non seulement à l'expérience des experts, mais aussi aux résultats numériques obtenus via la résolution des problèmes d'optimisation. Dans les applications réelles, les problèmes d'optimisation sont souvent de grande dimension faisant intervenir plusieurs paramètres. Par conséquent, le temps de calcul des méthodes d'optimisation exacte explose. De plus, le développement des méthodes d'optimisation exacte exige des informations sur la forme et la nature de la ou les fonction(s) objectif(s) à optimiser, ce qui n'est pas toujours possible.

Dans ce contexte, les métaheuristiques s'avèrent une alternative intéressante dans la mesure où d'une part, elles sacrifient la complétude de la résolution au profit de l'efficacité et du temps de calcul et d'autre part elle s'appliquent à un très large panel de problèmes.

La plupart des métaheuristiques sont d'inspiration biologique, physique ou génétique. Parmi elles, nous pouvons citer les algorithmes génétiques (évolutionnaires), le recuit simulé, la recherche tabou, l'optimisation par essaim particulaires et l'optimisation par colonie de fourmis. Ces algorithmes se déclinent en

général en deux types selon la nature du problème d'optimisation traité : mono-objectif ou multiobjectif.

En effet, dans certains domaines critiques de l'industrie, l'optimisation consiste à optimiser non seulement un seul critère mais plutôt simultanément plusieurs critères qui sont parfois conflictuels. La complexité et l'automatisation des équipements employés dans les milieux industriels exigent donc une gestion de maintenance accrue afin de garantir un niveau de service optimal en termes de fiabilité et de disponibilité. Toutefois, cette garantie d'optimalité a un coût très élevé, ce qui est contraignant. En pratique, cela se traduit par la recherche d'une politique de maintenance réalisant un compromis acceptable entre la disponibilité, la sécurité du système et de son environnement et les coûts associés à son entretien...

Dans son objectif de proposer une démarche de résolution d'un problème d'optimisation de la maintenance préventive, ce manuscrit de thèse fournit une méthodologie de résolution du problème d'optimisation des politiques de maintenance préventive systématique appliquée dans le domaine ferroviaire à la prévention des ruptures de rails. Le raisonnement de cette méthodologie s'organise autour de trois étapes principales :

1. Modélisation de l'évolution des comportements des états des composants constituant le système, i.e, connaître les mécanismes de dégradation des composants et formalisation des opérations de maintenance.
2. Formalisation d'un modèle d'évaluation de politiques de maintenance tenant compte aussi bien du facteur sûreté de fonctionnement du système que du facteur économique conséquent aux procédures de gestion de la maintenance (coûts de réparation, de diagnostic, d'indisponibilité).
3. Optimisation des paramètres de configuration des politiques de maintenance préventive systématique afin d'optimiser un ou plusieurs critères. Ces critères sont définis sur la base du modèle d'évaluation des politiques de maintenance proposé dans l'étape précédente.

Ces travaux de thèse ont été préparés au sein du Laboratoire Génie des Réseaux de Transports Terrestres et Informatique Avancée de l'institut français des sciences et technologies des transport, de l'aménagement et des réseaux (IFSTTAR), sous la direction de M. Patrice Aknin, directeur de recherche (IFSTTAR) et la co-direction de M. Patrick SIARRY Professeur des Universités (UPEC).

Le manuscrit se divise principalement en trois parties.

La première, réunissant les deux premiers chapitres, est consacrée à l'étude des approches de modélisation de la dégradation et d'évaluation des politiques de maintenance. La deuxième partie rapporte la contribution à l'application des métaheuristiques à l'optimisation mono-objectif et multiobjectif de la maintenance



préventive. Elle fait l'objet des chapitres 3 et 4. Enfin, la dernière partie illustre l'application des algorithmes proposés pour la résolution du problème d'optimisation de la maintenance pour la prévention des ruptures de rails. Elle concerne le chapitre 5.

Le premier chapitre présente une étude des méthodes de modélisation de la dégradation. L'accent a été mis sur l'approche de modélisation par modèles graphiques probabilistes, notamment les modèles graphiques de durée. C'est en se basant sur ce dernier que nous proposons un modèle d'évaluation des politiques de maintenance.

Le deuxième chapitre porte sur l'évaluation des politiques de maintenance. Les développements présentés dans ce chapitre ont abouti à la formalisation d'une fonction d'utilité. Cette dernière permet la quantification des stratégies de maintenance complexes appliquées à des systèmes multi-composants. Elle doit prendre en compte, à la fois, des contraintes économiques et des exigences de sûreté de fonctionnement.

Le troisième chapitre est consacré à la présentation d'un état de l'art des métaheuristiques les plus récentes. Une présentation non-exhaustive des approches générales de résolution de l'optimisation a été effectuée. Par la suite, nous définissons le problème d'optimisation, son vocabulaire et les notions relatives. Nous finissons ce chapitre par la description de différents types de métaheuristiques.

Le quatrième chapitre se compose essentiellement de deux parties. La première concerne l'optimisation mono-objectif. Des métaheuristiques mono-objectifs adaptées aux problèmes d'optimisation de la maintenance préventive systématique ont été développées. Le premier algorithme proposé est un algorithme génétique se déclinant en deux variantes. Les opérateurs de reproduction et les techniques de diversification ont été adaptés dans chaque algorithme afin de guider la recherche des solutions optimales au gré du type de la politique de maintenance préventive à optimiser. Le deuxième algorithme est un recuit simulé modifié. Ce dernier se distingue par sa technique hybride de recherche de voisinage.

Dans la deuxième partie du chapitre quatre l'optimisation multiobjectif a été abordée. Deux métaheuristiques multiobjectif sont proposées. Dans le premier algorithme qui est inspiré de l'algorithme ev-MOGA, la subdivision de l'espace des objectifs est dynamique. Les opérateurs de reproduction ont été à leur tour adaptés afin de mieux tenir compte du contexte de l'optimisation et de l'anatomie de la solution candidate. Le second algorithme, le NSGAI-modifié, définit une nouvelle procédure d'élimination des solutions non-dominées.

Le chapitre 5, présente l'application de la méthodologie de résolution de l'optimisation de la maintenance d'un système, à un problème réel de maintenance

d'une voie ferée et plus particulièrement du rail. Les différentes étapes de cette méthodologie sont décrites en commençant par la représentation et la modélisation de la dégradation de la voie puis l'évaluation de la maintenance et enfin l'optimisation des paramètres de la stratégie de maintenance déployée.

# Modélisation de la dynamique de dégradation des systèmes complexes

---

## Sommaire

1.1 Introduction . . . . .	5
1.2 Modèles à dégradation continue . . . . .	6
1.3 Modèles à dégradation discrète . . . . .	15
1.4 Modèles graphiques probabilistes temporels . . . . .	20
1.5 Conclusion . . . . .	37

---

## 1.1 Introduction

La dégradation d'un système se traduit par l'évolution progressive et parfois irréversible de son état ce qui conduit à la diminution des marges de sûreté nécessaires pour assurer son bon fonctionnement. La modélisation de la dégradation des systèmes est capitale pour la maîtrise des risques industriels, pour la gestion de leur durée de vie, pour l'optimisation des opérations d'inspection, de maintenance et de réparation, ainsi que pour le choix des engagements des constructeurs sur la garantie de leurs équipements.

Toutefois, le manque de données, les incertitudes sur les propriétés physiques des composants du système étudié et les variations des conditions d'exploitation conduisent à des incertitudes plus ou moins importantes qui se répercutent inmanquablement sur les méthodes d'évaluation de la dégradation réelle. Ceci rend difficile la prise de décision quant à une intervention appropriée. Cette situation est aggravée lorsque les enjeux sont importants en terme de sécurité, les données rares, l'observation courante éloignée de la prévision et les conditions d'exploitation ne sont pas bien contrôlées. Ce constat ne fait qu'accroître l'intérêt de disposer d'une méthodologie de modélisation de la dégradation des systèmes complexes.

## 1.2 Modèles à dégradation continue

Les modèles à dégradation continue tentent de modéliser l'évolution au cours du temps de la dégradation. Pour les systèmes croissants non auto-réparable, les états sont alors les résultats des réalisations d'un processus stochastiques à trajectoires monotones croissantes. Pour une estimation pertinente, il est nécessaire donc de connaître la loi d'accroissement de dégradation entre deux instants consécutifs pour pouvoir estimer le niveau de dégradation à toute date [Park 1988] (voir figure 1.1).

Selon [Barlow 1987], la dégradation est généralement supposée être un processus

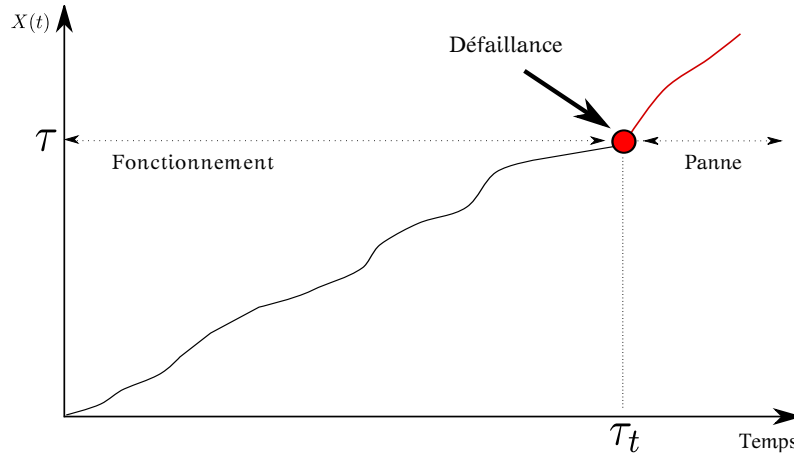


FIGURE 1.1 – Représentation d'un exemple de trajectoire d'états d'un modèle à dégradation continue

markovien. La dégradation à un instant  $t$  donné ne dépend que du niveau de dégradation à l'instant  $t - \delta t$  précédent, et de de l'intervalle de temps  $\delta t$  entre les deux instants. Cette hypothèse est justifiée si la seule information disponible sur l'état du système est l'accroissement de la dégradation du système entre ces deux instants [Castanier 2001].

Pour des systèmes à dégradation progressive, le taux de dégradation conditionnel à l'état courant  $\lambda_x(t)$  a été défini dans [Deloux 2008] de manière analogue au taux de défaillance  $\lambda(t)$  :

$$\lambda(t) = \lim_{\delta t \rightarrow 0} \frac{P(\tau_t \leq t + \delta t) \mid \tau_t > t)}{\delta t} \quad (1.1)$$

$$\lambda_x(t) = \lim_{\delta t \rightarrow 0} \frac{P(X(t + \delta t) \geq \tau \mid X(t) = x)}{\delta t} \quad (1.2)$$

tel que,  $x$ , le niveau courant de dégradation, est strictement inférieur au seuil de défaillance  $\tau$ .  $X(t)$  représente la dégradation sur un intervalle de temps de

longueur  $t$ .

Au vu, des propriétés de stationnarité et d'indépendance des accroissements, Les processus de Lévy [Asmussen 2003] conviennent pour la modélisation de la dégradation. La classe des processus de Lévy<sup>1</sup> réunit des processus largement étudiés dans la littérature [Ross 1995] : le processus gamma, le processus de Poisson composé et le processus de Weiner. Chacun de ces processus à temps continu a des particularités qui rendent son choix plus approprié dans certains cas.

L'apprentissage des modèles de dégradation peut se baser sur les instants de défaillance ou/et les mesures de dégradation. Le processus permettant l'observation des instants de défaillance peut être onéreux à mettre en place et souvent très long à réaliser (pour des systèmes très fiables) si les essais ne se font pas de manière accélérée. Par conséquent, l'utilisation de l'estimation des mesures de dégradation avant défaillance devient intéressante surtout si on considère plutôt les mesures des accroissements de dégradation qui sont indépendants. Cependant, ces deux types d'observations peuvent être utilisées conjointement comme dans [Nelson 1990].

### 1.2.1 Le processus de Wiener

Le processus de Wiener [Revuz 2004, Liao 2006, Nikulin 2006, Baussaron 2010] est souvent utilisé pour modéliser et rendre compte du mouvement brownien (MB), le "bruit", l'aléa pur, l'erreur de mesure physique...

Historiquement, il a été introduit pour la modélisation du mouvement irrégulier de particules de pollen en suspension dans l'eau, observées par Robert Brown. En 1920, Norbert Wiener a été le premier à proposer une définition mathématique de ce type de processus. Le processus de Wiener est souvent utilisé en mathématique appliquée, en physique et en économie : description du comportement thermodynamique des gaz, des phénomènes de diffusion, des modèles mathématiques de la finance, équations différentielles stochastiques .

**Définition 1.2.1** *On considère un espace de probabilité  $(\Omega, \mathcal{F}, \mathcal{P})$ . Un processus  $(B_t)_{t \geq 0}$  est un processus de Wiener standard s'il satisfait les propriétés suivantes :*

1.  $B_0(\omega) = 0$  pour tout  $\omega \in \Omega$
2.  $t \rightarrow (B_t(\omega))$  est une fonction continue,  $\forall \omega \in \Omega$
3. Les accroissements de  $(B_t)_{t \geq 0}$  sont indépendants et pour tout  $t \geq 0$  et  $h > 0$   $B_{t+h} - B_t$  suit la loi normale  $\mathcal{N}(0, h)$ .

---

1. un processus de Lévy est un processus stochastique, dont les accroissements sont stationnaires et indépendants

De par cette définition, le processus de Wiener permet d'effectuer et de simplifier certains calculs. Il est aussi utile à la modélisation parce qu'il hérite des propriétés des processus de Markov; il peut aussi être utilisé comme brique élémentaire pour la constructions de processus plus élaborés tel que le processus d'Ornstein-Uhlenbeck (qui peut représenter par exemple la vitesse d'une particule dans un milieu visqueux, soumise à des variations désordonnées dues aux chocs des molécules) ou bien le processus de Wiener avec dérive [Doksum 1992, Doksum 1995, Padgett 2004, Whitmore 1998].

**Définition 1.2.2 (Mouvement brownien avec dérive)** Soit  $(B_t)_{t \geq 0}$  un mouvement brownien standard. Nous appelons mouvement brownien avec dérive le processus  $(W_t)_{t \geq 0}$  vérifiant :

$$\begin{cases} W_0 = 0 & t = 0 \\ W_t = \sigma B_t + \mu t & \forall t > 0 \end{cases} \quad (1.3)$$

où  $\mu$  est une constante réelle appelée drift et  $\sigma^2$  ( $\sigma \geq 0$ ) le coefficient de diffusion. La variable aléatoire  $W_t$  est une v.a gaussienne d'espérance  $\mu t$  et de variance  $\sigma^2 t$ .

Le processus se comporte comme un mouvement brownien mais avec une tendance, positive ou négative selon la valeur du coefficient  $\mu$ . Ce processus a été utilisé par exemple pour modéliser l'évolution de l'espérance de l'état d'un système de freinage et illustrer la politique d'unique inspection pour l'entretien des plaquettes de frein d'une automobile [Crowder 2007]. Dans cette situation, il est très important de résoudre des problèmes liés au temps d'atteinte d'une barrière fixe comme le dépassement de la fiabilité d'un seuil sécuritaire qui correspond à une panne. Donc le système est considéré en panne dès que le processus  $W_t$  atteint une valeur seuil  $a \in \mathcal{R}$ .

Ainsi, le premier temps de passage  $T_a$  ( temps d'atteinte) d'une valeur  $a > 0$ , appelée aussi seuil de défaillance, est défini ci-dessous. Nous supposons que le coefficient de drift  $\mu$  est strictement positif.

$$T_a = \inf\{t > 0 : W_t \geq a\} \quad (1.4)$$

Le mouvement brownien à un temps fixé  $t > 0$  suit une loi normale  $\mathcal{N}(\mu t, \sigma^2 t)$ ; à l'inverse, le premier temps de passage, en lequel le mouvement brownien atteint un niveau donnée est étroitement lié a une distribution gaussienne inverse [Folks 1978]  $IG(a/\mu, a^2/\sigma^2)$ .

La fonction de densité de  $T_a$  est donnée par :

$$g_a(t) = \frac{a}{\sigma\sqrt{2\pi t}} \exp\left(-\frac{(\mu t - a)^2}{2\sigma^2 t}\right), \quad t > 0 \quad (1.5)$$

et la fonction de répartition de  $T_a$  est :

$$G_a(t) = \Phi\left(\frac{-a + \mu t}{\sigma\sqrt{t}}\right) + e^{2\mu a/\sigma^2} \Phi\left(\frac{-a - \mu t}{\sigma\sqrt{t}}\right), \quad t \geq 0 \quad (1.6)$$

Dans ce modèle, les accroissements ne sont pas nécessairement positifs ce qui peut être inadapté à la modélisation de certains phénomènes comme la modélisation de phénomènes de dégradation monotones (propagation d'une fissure, usure d'une pièce ...). Pour un tel modèle, à l'évidence, les accroissements ne sont pas nécessairement positifs, ce qui peut dans certains cas être incompatible avec le phénomène modélisé ou mesuré. Par conséquent ce modèle peut être inadapté à la modélisation de phénomènes de dégradation monotones tels que la propagation d'une fissure, ou encore l'usure d'une pièce.

**Exemple 1.2.1** Soit un système multi-composants, multi-états telqu'un système mécanique de distribution de fluide (SMDF). Le système est composé sommairement de deux unités A et B, montées en série, initialement neuves.

On suppose que la dégradation de chacune des unités du système est modélisée

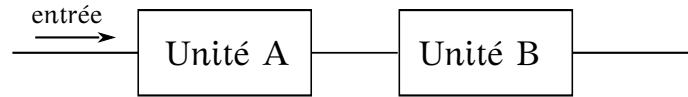


FIGURE 1.2 – Système à deux unités

par un processus de Wiener avec tendance. Il s'agit ici d'estimer les paramètres des processus de dégradations des unités A et B.

Soit  $W$  un processus de Wiener à tendance linéaire et qui modélise la dynamique de dégradation de l'unité A. Soit  $a > 0$  le seuil de défaillance.

En observant le temps d'atteinte de la valeur  $a$  pour  $m$  processus de Wiener avec tendances indépendantes, on a  $w_1, \dots, w_m$  variables aléatoires. La distribution des instants de défaillance est une loi gaussienne inverse  $IG(\eta, \lambda)$  Pour chaque unité,  $m$  valeurs de dégradations sont observées  $w_j = W(t_j)$  où  $0 < t_1 < \dots < t_m$  sont les instants d'observation  $i$ . Les méthodes d'estimation des paramètres classiques comme la méthode du maximum de vraisemblance, les estimateurs empiriques et la méthode des moments peuvent être utilisées. Nous utilisons la méthode du maximum de vraisemblance parce qu'elle présente plusieurs avantages :

- Dans de nombreux cas pratiques, il est plus facile d'optimiser  $\ln(\mathcal{L}(w_1, \dots, w_m; \theta))$  et souvent l'estimateur  $\hat{\theta}$  résout le système

$$\frac{\partial \ln \mathcal{L}}{\partial \theta}(w_1, \dots, w_m; \hat{\theta}) \quad (1.7)$$

- L'écart-type des estimations peut être estimé lorsque la longueur  $m$  de la série chronologique est grande. En effet, la matrice de variances-covariances de  $\hat{\theta}$  est approchée par :

$$\hat{\Sigma} = -1 / \left( \frac{\partial^2 \ln(\mathcal{L})}{\partial \theta_i \partial \theta_j}(w_1, \dots, w_m; \hat{\theta}) \right) \quad (1.8)$$

- Il est possible de calculer des intervalles de confiance et d'exécuter des tests d'hypothèses puisque la loi  $\hat{\theta} - \theta$  est approximativement une loi normale d'espérance nulle et de variance  $\hat{\Sigma}$  lorsque  $m$  est grand.
- Si la quantité que l'on souhaite estimer est une transformation des paramètres,  $\alpha = g(\theta)$ , alors  $\hat{\alpha} = g(\hat{\theta})$  et l'estimateur de la variance de l'estimateur  $\hat{\alpha}$  de  $\alpha$  est

$$\widehat{\text{var}}(\hat{\alpha}) = g'(\hat{\theta}) \widehat{\text{var}}(\hat{\theta})$$

Les paramètres  $\eta$  et  $\lambda$  sont obtenus par maximisation de la log-vraisemblance :

$$\begin{aligned} \ln(\mathcal{L}(\eta, \lambda)) &= \ln(\lambda^{m/2} \prod_{i=1}^m (2\pi w_i)^{-1/2} \exp(-\frac{\lambda(w_i - \eta)^2}{2\eta^2 w_i})) \\ &= \frac{n}{2} \ln \lambda - \frac{1}{2} \sum_{i=1}^m (2\pi w_i^3) - \frac{\lambda}{2\eta^2} \sum_{i=1}^m \frac{(w_i - \eta)^2}{w_i} \end{aligned} \quad (1.9)$$

$$\frac{\partial}{\partial \eta} \mathcal{L}(\eta, \lambda) = \frac{\lambda}{\eta^3} \sum_{i=1}^m \frac{(w_i - \eta)^2}{w_i} + \frac{\lambda}{\eta^2} \sum_{i=1}^m \frac{(w_i - \eta)}{w_i} = 0 \quad (1.10)$$

alors

$$\frac{1}{\eta} \sum_{i=1}^m w_i + \eta \sum_{i=1}^m \frac{1}{w_i} - 2m = -m + \eta \sum_{i=1}^m \frac{1}{w_i} \quad (1.11)$$

d'où

$$\hat{\eta} = \frac{1}{m} \sum_{i=1}^m w_i = w \quad (1.12)$$

En fixant  $m = w$ ,  $\lambda$  peut être obtenu en résolvant l'équation suivante :

$$\begin{aligned} \frac{\partial}{\partial \lambda} \mathcal{L} &= \frac{m}{2\lambda} - \frac{1}{2w^2} \sum_{i=1}^m (w_i + \frac{w^2}{w_i} - 2w) \\ &= \frac{m}{2\lambda} - \frac{1}{2w^2} \sum_{i=1}^m w_i - \frac{1}{2} \sum_{i=1}^m \frac{1}{w_i} + \frac{m}{w} \end{aligned} \quad (1.13)$$

d'où

$$\hat{\lambda} = \frac{1}{m} \sum_{i=1}^m \left( \frac{1}{w_i} - \frac{1}{w} \right) \quad (1.14)$$



### 1.2.2 Le processus Gamma

Le processus gamma est un processus de chocs à temps continu et à accroissements indépendants et stationnaires<sup>2</sup> qui décrit des trajectoires de dégradation constante. Ses trajectoires sont continues à droite et admettent des limites à gauche.

Le processus gamma est fréquemment utilisé dans la littérature pour modéliser les mécanismes de vieillissement qui sont par nature lents, continus, progressifs et non résorbables [Abdel-Hameed 1975]. Son utilisation dans cette optique a été justifiée dans plusieurs travaux, notamment dans [Noortwijk 2009]. Ce processus est essentiellement adapté pour modéliser la phase de propagation des dégradations monotones et croissantes dans le temps comme la corrosion, la propagation de fissures ou l'érosion. Il convient aussi très bien pour la modélisation de la variabilité temporelle de détérioration et pour la détermination des décisions de maintenance et d'inspection optimales [Castanier 2005], [Crowder 2007], [Dieulle 2003], [Kong 1997]. Ce choix de modélisation suppose nécessairement que l'état du système à travers le temps ne s'améliore pas et qu'il est impossible au système de retrouver son état initial ou un état précédent sans intervention de maintenance. Plusieurs travaux portant sur la maintenance et la modélisation de la dégradation ont été menés avec le processus de gamma. A titre d'exemple, nous citons d'une part les travaux de [Crowder 2007], [Castanier 2005] et de [Dieulle 2003], appliqués respectivement, pour la recherche de politiques de maintenance préventive, la recherche de politiques de maintenance conditionnelle à fréquence d'inspections non périodiques et la planification de maintenances séquentielles. D'autre part, nous rappelons aussi les travaux de [Grall 2006] et de [Liao 2006] qui illustrent le processus de gamma dans la recherche d'un seuil de défaillance pour une politique de maintenance de systèmes en surveillance continue.

**Définition 1.2.3** *Le processus  $(Y_t)_{t \geq 0}$  est un processus gamma homogène et stationnaire si :*

1.  $Y_0 = 0$ ,
2. Pour tout  $n \geq 1$  et tout  $0 \leq t_1 < t_2 < \dots < t_n$ , les accroissements  $Y_{t_1}, Y_{t_2} - Y_{t_1}, \dots, Y_{t_n} - Y_{t_{n-1}}$  sont indépendants ;
3. Pour tout  $\Delta t > 0$ ,  $t > 0$ , la loi de l'accroissement  $Y_{t+\Delta t} - Y_t$  est une loi gamma  $G(\alpha \Delta t, \beta)$  où  $\alpha, \beta \in \mathcal{R}_+^*$ .

*On dit que  $\alpha$  est un paramètre de forme et  $\beta$  un paramètre d'échelle.*

---

2. La loi des accroissements (propriétés statistiques caractérisées par des espérances mathématiques) est indépendante de l'âge du processus

Dans le cas d'un processus gamma non homogène,  $\alpha$  est alors une fonction donnée, monotone et croissante [Abdel-Hameed 1975], [Bagdonavicius 2001].

Pour  $t \geq 0$ ,  $Y_t$  suit la loi  $Ga(\alpha t, \beta)$  homogène, de moyenne  $\mu_t = \alpha t / \beta$ , de variance  $\sigma_t^2 = \alpha t / \beta^2$  et de densité :

$$f_{\alpha t, \beta}(y) = \frac{\beta^{\alpha t} y^{\alpha t - 1} e^{-\beta y}}{\Gamma(\alpha t)} \quad t > 0, y \geq 0 \quad (1.15)$$

La fonction  $\Gamma$  est définie par :

$$\Gamma(z) = \int_0^{+\infty} y^{z-1} e^{-y} dy \quad (1.16)$$

Le taux de défaillance est défini par :

$$\lambda(t) = \frac{\beta^{\alpha t} t^{\alpha t - 1} e^{-\beta t}}{\int_t^{\infty} \Gamma(\alpha t) f_{\alpha t, \beta}(y) dy} \quad (1.17)$$

Toutes les caractéristiques situées ci-dessus rendent l'utilisation de ce processus intéressante dans la modélisation des mécanismes de dégradation, mais ce n'est pas sans difficultés. En effet les données disponibles pour la modélisation avec le processus gamma sont généralement les données de mesure de l'état de dégradation d'un composant et les dates des inspections. Néanmoins, l'obtention de ces données en bonne et due forme n'est pas garantie car en réalité les inspections sont effectuées de façon plus ou moins régulière et les défauts ne sont pas toujours correctement identifiés.

Dans le domaine de la maintenance, le processus de Gamma est souvent utilisé tout simplement comme un outil de modélisation à paramètres connus ou faciles à estimer dans le but d'optimiser la prise de décisions.

**Exemple 1.2.2** Reprenons l'exemple du système à deux unités. On suppose que ces deux unités se dégradent chacune suivant un processus de gamma stationnaire homogène  $Y_i$ ,  $i \in \{1, 2\}$ .

Le but est alors d'estimer les paramètres  $(\alpha, \beta)$  de chaque processus Gamma  $Y_i$ .

Les données pour l'estimation de ces paramètres peuvent représenter  $m$  valeurs de dégradation observées pour une unique trajectoire :

$y_j = Y(t_j)$  sont les observations et  $t_j, \forall j \in \{0, \dots, m\}$ , sont les instants d'inspection associés.

On considère alors les accroissements de dégradation  $\Delta y_j = Y(t_j) - Y(t_{j-1}), j \in \{1 \dots m\}$ . Les incréments de temps sont tous égaux  $\Delta t = t_j - t_{j-1}, j \in \{1 \dots m\}$ .

Nous utilisons pour cet exemple, la méthode du maximum de vraisemblance.

La fonction de vraisemblance des accroissements  $\Delta y_j, j \in \{1 \dots m\}$  est donnée par :

$$\begin{aligned} \mathcal{L}(\alpha, \beta) &= \prod_{j=1}^m f_{\alpha\Delta t, \beta}(\Delta y_j) \\ &= \prod_{j=1}^m \frac{\beta^{\alpha\Delta t}}{\Gamma(\alpha\Delta t)} (\Delta y_j)^{\alpha\Delta t-1} e^{-\beta\Delta y_j} \end{aligned} \quad (1.18)$$

et la log-vraisemblance est :

$$\ln(\mathcal{L}(\alpha, \beta)) = \sum_{j=1}^m (\alpha\Delta t \ln(\beta) - \ln(\Gamma(\alpha\Delta t)) + (\alpha\Delta t - 1)\ln(\Delta y_j) - \beta\Delta y_j) \quad (1.19)$$

Or

$$\frac{\partial \alpha}{\alpha} (\ln \mathcal{L}(\alpha, \beta)) = \sum_{j=1}^m (\Delta t \ln(\beta) - \Delta t \times \psi(\alpha\Delta t) + \Delta t \ln(\Delta y_j)) \quad (1.20)$$

$$\frac{\partial \beta}{\beta} (\ln \mathcal{L}(\alpha, \beta)) = \sum_{j=1}^m \left( \frac{\alpha\Delta t}{\beta} - \Delta y_j \right) \quad (1.21)$$

où  $\psi$  est la fonction Digamma :

$$\psi(x) = \frac{\Gamma'(x)}{\Gamma(x)} \quad (1.22)$$

Par conséquent les estimateurs  $\hat{\alpha}$  et  $\hat{\beta}$  de  $\alpha$  et  $\beta$  sont obtenus en résolvant les équations suivantes :

$$m \ln(\hat{\alpha} \frac{n\Delta t}{\sum_{j=1}^m \Delta y_j}) + \Delta t \sum_{j=1}^m (\ln(\Delta y_j) - \psi(\hat{\alpha}\Delta t)) = 0 \quad (1.23)$$

La première équation (1.19) doit être résolue par une méthode numérique telle que la méthode de Newton.

### 1.2.3 Le processus de Poisson composé

Étant un processus de Lévy, le processus de Poisson composé est un processus continu à droite, limité à gauche et à accroissements indépendants et stationnaire. Il correspond, alors, à un processus constant par morceaux qui saute aux instants de sauts d'un processus de Poisson standard, et dont les tailles de sauts

sont des variables indépendantes et identiquement distribuées (i.i.d) d'une loi donnée [Rémi 2010].

Considérons un processus de Poisson  $(N_t)_{t \geq 0}$  d'intensité  $\lambda > 0$ , de temps de sauts  $\tau_n$ , et une suite  $(Z_n)_{n \in \mathbb{N}^*}$  de v.a. tel que :

1.  $(Z_n)$  est une suite de v.a i.i.d à valeurs dans  $\mathbb{R}^+$  et de loi  $\nu_z$ .
2.  $(N_t)_{t \geq 0}$  est un processus de Poisson de paramètre  $\lambda$  indépendant de la suite  $(Z_n)_{n \in \mathbb{N}^*}$ .

On pose

$$X_t = \sum_{k=1}^{N_t} Z_k = \sum_{k=1}^{+\infty} Z_k 1_{[0,t]}(\tau_k) \quad (1.24)$$

**Définition 1.2.4** *Le processus  $(X_t)_{t \geq 0}$  est un processus de Poisson composé d'intensité  $\lambda$  et de distribution de sauts  $\nu_z$ .*

*$(X_t)_{t \geq 0}$  est alors à accroissements indépendants et stationnaires.*

Le processus de poisson a été largement utilisé pour modéliser les phénomènes de temps d'arrivée aléatoires (sinistres, clients, accident, trafic,...), mais aussi pour l'étude de fiabilité et de modélisation de dégradation. Citons par exemple [Lam 2003] qui l'utilise pour modéliser la dégradation d'un système qui évolue dans un environnement aléatoire afin de justifier leur modèle de maintenance basé sur un processus géométrique. Dans [Barbour 1995] les auteurs démontrent que l'utilisation du processus de poisson composé pour la mesure de fiabilité est efficace et peut l'être aussi pour des systèmes plus compliqué que celui étudié dans leur étude et également, que l'utilisation de ce dernier pour l'approximation de la fiabilité est souvent meilleure qu'avec une approche de processus de poisson local.

Il apparaît clairement que ces trois processus stochastiques à temps continu : processus de Wiener, processus de Gamma et le processus de Poisson composé partagent les mêmes objectifs pour une modélisation fidèle de la dynamique de dégradation de systèmes.

### 1.3 Modèles à dégradation discrète

Dans la plupart des situations physiques, le temps est une variable continue. Les phénomènes comme le mouvement d'une particule, l'apparition d'une panne ou l'usure d'un composant mécanique sont donc des variables aléatoires qui peuvent occuper n'importe quelle position temporelle.

Cependant, une discrétisation temporelle du temps simplifierait plus les calculs à

condition que la description du modèle soit correcte et que le passage du temps discret au temps continu n'altère pas les résultats essentiels du problème. Il suffit alors de prendre une limite de pas de temps de discrétisation adéquate au problème.

Dans les modèles à dégradation discrète l'ensemble possible des états du système est fini. Chaque état désigne un niveau de dégradation du système. Ceci suppose que le système se dégrade suivant un certain nombre d'états de dégradation que l'on peut qualifier de transitoires et qui sont généralement classés par ordre croissant de gravité allant du système neuf jusqu'à la défaillance.

Le système garde un même état de dégradation pendant une durée de temps qui varie selon plusieurs contextes (nature de la dégradation courante, type de système, ...). Un état de dégradation pourrait être représenté donc sous forme d'un palier de dégradation. Entre deux paliers de dégradation successifs, l'état du système ne change pas. En revanche, en l'absence d'interventions de maintenance, ce dernier évolue toujours vers des états de plus en plus dégradés jusqu'à la défaillance totale du système.

Dans tous ces modèles, il est également possible de prendre en compte des variables influant sur la vitesse de dégradation des systèmes (effets sur la tendance, la variance, changement de l'échelle de temps, ...). Parmi les modèles à dégradation discrète les plus répandus il y a le processus de poisson et les processus markoviens ou semi-markoviens. Ces modèles peuvent être utilisés pour définir une politique de maintenance optimale selon un critère donné. Par exemple, [Feldman 1977] modélise l'instant du saut (choc) par un processus semi-markovien, en vue de définir une politique de remplacement optimale, minimisant un certain coût. Dans [Castanier 2001], les processus markoviens sont utilisés pour modéliser le processus d'évolution d'un système à dégradation graduelle.

La transition d'un état  $i$  à un autre état  $j$  du système est gérée par une matrice de transition. Le taux de transition est constant dans le cas markovien. Il est fonction du temps dans le cas non markovien. L'estimation de cette matrice de transition est d'autant plus délicate qu'elle nécessite généralement un grand nombre de données de retours d'expérience.

### 1.3.1 Processus de poisson

Le processus de Poisson est un processus fréquemment utilisé pour modéliser les instants auxquels se produisent des phénomènes aléatoires, comme l'arrivée d'appels téléphoniques ou encore les instants de pannes d'un équipement.

Ce modèle doit décrire le nombre d'occurrence d'un événement aléatoire répétitif, pendant un intervalle de temps donné. Il s'agit alors d'un processus de comptage.

Soit  $\{N_t, t \geq 0\}$  est un processus de comptage.  $N_t$  représente le nombre d'événements qui se sont produits dans l'intervalle de temps  $[0, t]$ .

**Définition 1.3.1** Soit  $(T_n)_{n \geq 1}$  une suite de variables aléatoires tel que  $(T_{n+1} - T_n)_{n \geq 1}$  sont indépendantes et identiquement distribuées selon une loi exponentielle de paramètre  $\lambda > 0$ . Les variables  $T_n$  représentent les temps d'occurrences tel que  $T_0 = 0$ . Un processus de Poisson  $(N_t)_{t \geq 0}$  de paramètre  $\lambda t$  est un processus de comptage associé à  $T_n$  tel que

$$(N_t)_{t \geq 0} = \sum_{n \geq 1} \mathbf{1}_{\{T_n \leq t\}} \quad (1.25)$$

$$P(N_t = n)_{n \in \mathbb{N}} = e^{-\lambda t} \frac{(\lambda t)^n}{n!} \quad (1.26)$$

En particulier on a :

$$\mathbb{E}[N_t] = \lambda t \quad (1.27)$$

$$(1.28)$$

Plaçons nous, maintenant, dans un contexte de fiabilité. Considérons un système  $S$  se dégradant au cours du temps, selon un taux de défaillance  $\lambda(t)$ . On suppose qu'initialement  $S$  est à l'état neuf et que lorsque une défaillance survient elle est réparée instantanément. Cette hypothèse suppose que la réparation est effectuée en dehors du temps de fonctionnement ou bien que la durée de réparation est tellement courte qu'elle devient négligeable par rapport au temps de bon fonctionnement. Après l'intervention de maintenance deux cas de figures sont distingués :

- le système est remis à neuf : il s'agit alors d'une réparation totale. Le taux de défaillance  $\lambda(t) = \lambda(0)$ .
- le système est remis en marche mais le taux de défaillance n'est pas modifié : la réparation effectuée est minimale. Autrement, pour tout  $n > 0$ , sachant que les  $(n - 1)$  premières défaillances sont survenues aux instants  $t_1, \dots, t_{n-1}$ , la durée de la  $n^{\text{ième}}$  période de bon fonctionnement a pour taux de défaillance la fonction  $h \rightarrow \lambda(t_{n-1} + h)$ . Ceci, implique que les instants successifs de défaillance se modélisent par un processus de Poisson d'intensité  $t \rightarrow \lambda(t)$ .

Dans la littérature le processus de poisson a été souvent utilisé pour la modélisation de la dégradation. Par exemple les auteurs [Castro 2009] utilisent deux processus de poisson non homogènes différents pour modéliser l'arrivée des défaillances réparables et non réparables. Sur la base de ce modèle de dégradation un calendrier de maintenance préventive est établi avant le remplacement totale du système.

Dans [Nakagawa 1989] et [Qian 2003] les auteurs utilisent le processus de poisson non homogène pour modéliser l'arrivée des défaillances (réparables et non réparables). Sur la base de ce modèle de dégradation un calendrier de maintenance minimale et de remplacement est établi. Aussi [Yeh 2009] utilisent un processus de poisson non homogène comme modèle de dégradation afin de trouver une politique de maintenance préventive optimale pour les équipements en location.

### 1.3.2 Chaînes de Markov

L'approche Markovienne est la plus utilisée pour le traitement probabiliste des systèmes dynamiques. Une chaîne de Markov à valeurs dans un espace fini  $E$  est un processus stochastique à temps discret qui vérifie la propriété de Markov. Cette propriété suppose que l'estimation de l'état du futur dépend seulement du présent indépendamment du passé.

Les chaînes de Markov reposent sur deux concepts de base : les états et les transitions. L'état d'un système est caractérisé par les différents états de ses constituants (par exemple opérationnels ou en défaut). Lorsque l'espace des états du système est un ensemble discret fini au cours du temps (par exemple passage de l'état opérationnel à un état de dégradation et d'un état dégradé ou panne à l'état opérationnel suite à une action de maintenance). Ces changements d'état sont appelés les transitions.

Les chaînes de Markov reposent sur l'hypothèse que les taux de transition d'un système (taux de défaillance et de réparation) sont constants et que le processus est sans mémoire (Markovien). Un processus stochastique  $X_t, t \in \mathcal{T}$  est une collection de variables aléatoires définies sur un espace probabilisé  $(\Omega, \mathcal{F}, \mathcal{P})$ . La variable  $X_t$  représente l'état du processus au temps  $t$  et l'ensemble de toutes les valeurs possibles pour cette variable est appelée l'espace des états du processus et sera noté  $\mathcal{E}$ .

Un processus stochastique dont l'ensemble des états  $\mathcal{E}$  est fini ou dénombrable est appelé une chaîne. Le processus est à temps discret lorsque l'ensemble  $\mathcal{T}$  est fini ou dénombrable.

Certains auteurs réservent la dénomination de "chaîne" aux processus de Markov à temps discret, et appellent alors processus de Markov à espace d'état dénombrable ce que nous avons appelé chaîne de Markov à temps continu. D'autres préfèrent réserver l'appellation "processus de Markov" aux processus à espace d'état non dénombrable vérifiant la propriété de Markov.

### 1.3.2.1 Définition

**Définition 1.3.2** *une chaîne de Markov à temps discret est un processus stochastique  $(X_n)_{n \geq 1}$  à temps discret, à espace d'états  $\mathcal{E}$  fini ou infini et vérifiant la propriété de Markov :*

$$P[X_{n+1} = j \mid X_n = i, X_{n-1} = x_{n-1}, \dots, X_0 = x_0] = P[X_{n+1} = j \mid X_n = i] \quad (1.29)$$

tel que  $(x_k)_{0 \leq k \leq n-1}, i, j \in \mathcal{E}$

**Définition 1.3.3** *Une chaîne de Markov à temps discret est dite homogène si, pour tout couple d'états  $(i, j)$  et pour tout instant  $n$  on a*

$$P[X_{n+1} = j \mid X_n = i] = P[X_{n+k} = j \mid X_{n+k-1} = i], \quad k \geq 0 \quad (1.30)$$

Par conséquent,

$$P[X_{n+1} = j \mid X_n = i] = P[X_1 = j \mid X_0 = i] = p_{i,j}, \quad \forall (i, j) \in \mathcal{E}^2 \quad (1.31)$$

$p_{i,j}$  représente la probabilité de transition initiale de  $i$  à  $j$ ; en d'autres termes c'est la probabilité conditionnelle que le système se retrouve dans l'état  $j$  à l'étape suivante sachant qu'il se trouve actuellement dans l'état  $i$ .

Pour un système à  $|\mathcal{E}|$  états, la matrice de transition  $P = (p_{i,j})$  sera de taille  $|\mathcal{E}| \times |\mathcal{E}|$ . Les lignes et colonnes de la matrice sont indexés par les éléments de  $\mathcal{E}$ .

De nombreux travaux ont utilisé les chaînes de Markov dans l'optimisation de la maintenance. Par exemple dans [Chan 2006] les auteurs se basent sur un processus de décision markovien pour conclure une politique de maintenance optimale. Aussi, dans [Soro 2010] les auteurs développent un modèle d'évaluation de disponibilité, de taux de production et de fiabilité d'un système dégradable et assujéti à des réparations minimales et à une maintenance préventive imparfaite. Le système est modélisé comme un processus de markov à temps continu.

### 1.3.3 Processus Semi-markovien

L'étude des chaînes semi-markoviennes permet d'étudier des processus dont le temps de séjour dans un état peut suivre une loi discrète quelconque (pas forcément exponentielle), tout en gardant une évolution markovienne entre les états successifs.

La structure d'un processus semi-markovien est identique à celle d'un processus de markov de sauts<sup>3</sup>, tel que les transitions sont réalisées suivant une chaîne de

3. un processus markovien de sauts est une chaîne de Markov en temps continu, à valeurs dans un ensemble fini ou dénombrable  $\mathcal{E}$ , qui évolue par sauts successifs à des instants aléatoires



Markov.

La seule différence entre ces deux processus vient du fait que la durée entre deux sauts pour un processus de markov de sauts suit une loi exponentielle alors que pour un processus semi-markovien elle peut suivre n'importe quelle distribution. Soit  $\mathcal{E} = 1, \dots, s$  l'ensemble des états du système à modéliser. On note  $S_n$  une suite croissante de variables positives :

$$0 = S_0 \leq S_1 \leq S_2 \leq \dots \leq S_n \leq \dots$$

tel que  $\tau_n = S_{n+1} - S_n$  et  $J_n$  est un processus à valeurs dans  $\mathcal{E}$ . Soit  $X_t$  l'état du processus à l'instant  $t$  où  $X_t = J_n$ , pour  $S_n \leq t \leq S_{n+1}$

**Définition 1.3.4** On dit que le processus  $(J, S) = (J_n, S_n)_{n \geq 0}$  est une chaîne de renouvellement markovien (homogène en temps) à valeurs dans  $\mathcal{E}$  et que  $X_n$  est la chaîne semi-markovienne (notée SMC) associée au processus  $(J, S)$  si,  $\forall n \in \mathbb{N}$  :

$$P(J_{n+1} = j, \tau_n = k | J_0, S_0, \dots, J_n, S_n) = P(J_{n+1} = j, \tau_n = k | J_n = i) = q_{ij}(k) \quad (1.32)$$

Cette condition signifie que l'évolution future d'un processus semi-markovien (temps de séjour dans l'état présent et état suivant) ne dépend que de l'état présent de celui-ci, et non des états antérieurs ou des temps de séjour en ces derniers.

$q_{ij}(k)$  s'interprète comme la probabilité de passer de  $i$  en  $j$  après un temps de séjour de  $k$  temps en  $i$ .

Soit  $\mathcal{M}_{\mathcal{E}}$  l'ensemble des matrices  $\mathcal{E} \times \mathcal{E}$ . Soit  $\mathcal{M}_{\mathcal{E}}(\mathbb{N})$  l'ensemble des applications de  $\mathbb{N}$  dans  $\mathcal{M}_{\mathcal{E}}$ . On peut introduire la définition suivante pour les matrices de transition semi-markoviennes :

**Définition 1.3.5**  $q \in \mathcal{M}_{\mathcal{E}}(\mathbb{N})$  est une matrice de transition semi-markovienne si :

- i)  $q_{ij}(k) \geq 0, \forall i, j \in \mathcal{E}, \forall k \in \mathbb{N}$
- ii)  $q_{ij}(0) = 0 \forall i, j \in \mathcal{E}$
- iii)  $\sum_{k \in \mathbb{N}} \sum_{j \in \mathcal{E}} q_{ij}(k) = 1 \forall i \in \mathcal{E}$

Si  $(J_n, S_n)$  est une chaîne de renouvellement markovien homogène,  $J_n$  est une chaîne de Markov de matrice de transition  $p_{ij} = \sum_{k \in \mathbb{N}} q_{ij}(k)$ .

**Définition 1.3.6**

$\forall i, j \in \mathcal{E}, \forall k \in \mathbb{N}$  :

(i) La loi conditionnelle de  $X_n$  :

$$f_{ij}(k) = P(\tau_n = k | J_n = i, J_{n+1} = j) = \begin{cases} \frac{q_{ij}(k)}{p_{ij}} & \text{si } p_{ij} \neq 0 \\ 1_{k=\text{Inf}} & \text{sinon} \end{cases} \quad (1.33)$$

(ii) La fonction de répartition conditionnelle de  $X_n$  :

$$F_{ij}(k) = \sum_{l=0}^k f_{ij}(l) \quad (1.34)$$

(iii) Loi du temps de séjour en  $i$  :

$$h_i(k) = P(\tau_n = k | J_n = i) = \sum_{j \in E} q_{ij}(k) \quad (1.35)$$

(vi) La fonction de répartition du temps de séjour en  $i$

$$H_i(k) = \sum_{l=0}^k h_i(l) \quad (1.36)$$

Pour une illustration de l'utilisation de ces modèles, les auteurs [Chen 2005] ont construit un processus de décision semi-markovien pour l'optimisation des stratégies de maintenance préventive conditionnelle. Ils ont présenté aussi une approche pour conjuguer l'optimisation des taux d'inspection et des politiques de maintenance. Plus récemment, l'étude de [Kim 2009] a représenté un modèle de réparation pour un système multi-états assujetti à des défaillances majeures et mineures.

## 1.4 Modèles graphiques probabilistes temporels

Récemment, de nouveaux travaux ont révélé que les Modèles Graphiques Probabilistes (MGP), i.e. les réseaux bayésiens et les réseaux bayésiens dynamiques, sont pertinents pour représenter des systèmes complexes et réaliser des études de fiabilité. Citons à titre d'exemple les travaux de [Langseth 2007] et [Boudali 2005]. L'inconvénient de cette approche vient de la contrainte imposée sur les temps de séjour dans les états du système nécessairement exponentiellement distribués. Cette contrainte peut être levée par l'utilisation de Modèles Graphiques de Durée (MGD), qui permettent de prendre en compte n'importe quelle loi de durée.

Les MGDs se basent sur les modèles à variables de durée introduits par [Murphy 2002], qui permettent de représenter l'évolution du comportement d'un système dynamique au cours du temps.

Dans les paragraphes suivants, nous présentons d'abord, les réseaux bayésiens dynamiques et puis les Modèles Graphiques de Durée (MGD) ainsi que leur version dédiée à la modélisation des temps de séjours [Donat 2010].

### 1.4.1 Réseaux bayésiens dynamiques

#### 1.4.1.1 Réseaux bayésiens statiques

Les réseaux bayésiens statiques (RB), initiés par [Pearl 1985], sont des modèles graphiques qui représentent les relations probabilisées entre un ensemble de variables. La figure 1.3 représente un exemple de réseau bayésien statique.

Ils reposent ainsi sur un formalisme basé sur les théories des probabilités et des graphes et permettent de représenter de façon intuitive et parcimonieuse la loi d'une suite de variables aléatoires  $\mathbf{X} = (X_1, \dots, X_n)$ . Formellement un RB (statique) noté  $M = (G, (L_n)_{1 \leq n \leq N})$  est défini par :

- L'ensemble des variables aléatoires observables  $\mathbf{X} = (X_1, \dots, X_n)$ .
- $G = (\mathbf{X}, E)$ , graphe orienté sans circuit où chaque nœuds est associé à une variable de  $\mathbf{X}$  prenant ses valeurs dans  $\chi$ . Soit  $i$  et  $j$  deux nœuds, l'arc orienté  $(i, j) \in E$  indique une relation de dépendance entre les v.a.  $X_i$  et  $X_j$ .
- $(L_n)_{1 \leq n \leq N}$ , suite de lois de probabilité conditionnelles (LPC) telle que chaque  $L_n$  représente la loi de probabilité de la v.a  $X_n$  conditionnellement à ses parents immédiats  $X_{pa_n}$ .  $pa_n$  désigne les indices des v.a. "parentes" de  $X_n$  dans  $G$ .

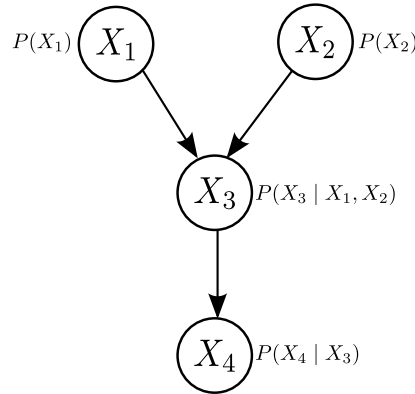


FIGURE 1.3 – Exemple de réseau bayésien statique

Les relations d'indépendance conditionnelle introduites par les arcs du graphe permettent de factoriser la distribution jointe de la suite de v.a.  $\mathbf{X}$  de la manière suivante :

$$P(\mathbf{X}) = \prod_{i=1}^N P(X_i | X_{pa_i}) \quad (1.37)$$

#### 1.4.1.2 Version dynamique

Les Réseaux Bayésiens Dynamiques (RBDs) [Dean 1989] sont un cas particulier de réseaux bayésiens classiques, capables de représenter l'évolution

de certaines variables dans le temps. Ils permettent ainsi de modéliser des systèmes dynamiques, au même titre que les Modèles de Markov Cachés (HMM) [Baum 1970, Rabiner 1986] ou les filtres de Kalman (KF) [Kalman 1960]. Ces modèles très différents partagent tous la même propriété de Markov : "Sachant le présent, le futur est indépendant du passé". Cependant la particularité des RBDs est de pouvoir prendre en considération les interactions locales au sein d'un état pour représenter de manière compacte le système.

Un RBD est composé d'une suite de tranches de temps, où chaque tranche contient un ensemble de variables représentant l'état de l'environnement pour ce laps de temps. Chaque tranche en soi est un réseau bayésien, et la même structure de réseau est répliquée au fur et à mesure que le temps avance.

La dynamique temporelle de l'environnement qui modélise le passage d'une tranche à l'autre est caractérisée par des matrices de transitions et représentée par des arcs reliant les différentes tranches entre elles. La figure ?? illustre un exemple de réseau bayésien dynamique sur 4 tranches de temps.

Un réseau bayésien dynamique (RBD) code la distribution jointe de probabilités d'un ensemble de variables  $(\mathbf{X}_t)_{1 \leq t \leq T} = (X_{1,t}, \dots, X_{N,t})_{1 \leq t \leq T}$  évoluant dans le temps sur une séquence de longueur  $T$ .  $\mathbf{X}_t$  représente l'état du système à un instant donné  $t$ .

Si on se fixe  $T$  pas de temps, le RBD peut être considéré comme un réseau bayésien (statique) déroulé avec  $T \times N$  variables.

En raison de la nature non déterministe des systèmes étudiés, on utilise des processus stochastiques pour les modéliser. On définit donc une distribution de probabilité jointe  $P((\mathbf{X}_t)_{1 \leq t \leq T})$  sur l'ensemble des états possibles qu'a pu prendre le système au cours du temps.

Du fait du nombre important d'états, et donc du nombre important de paramètres pour représenter cette loi, il est nécessaire de profiter de certaines propriétés des RBs pour la calculer en pratique. En effet en admettant que le processus est Markovien et invariant dans le temps, il est possible de représenter la loi jointe de manière compacte à l'aide d'une loi a priori  $P(\mathbf{X}_1)$ , qui représente la distribution sur l'état initial du système  $\mathbf{X}_1$  et un modèle de transition  $P(\mathbf{X}_t | \mathbf{X}_{t-1})$ ,  $t > 1$  qui donne la probabilité que le système passe dans l'état  $\mathbf{X}_t$  sachant qu'il était dans l'état  $\mathbf{X}_{t-1}$  au pas précédent.

En utilisant la propriété de factorisation des RBs (1.37), la densité jointe de probabilités s'écrit alors comme suit :

$$P((\mathbf{X}_t)_{1 \leq t \leq T}) = \prod_{t=1}^T \prod_{n=1}^N P(X_{n,t} | X_{pa_{n,t}}). \quad (1.38)$$

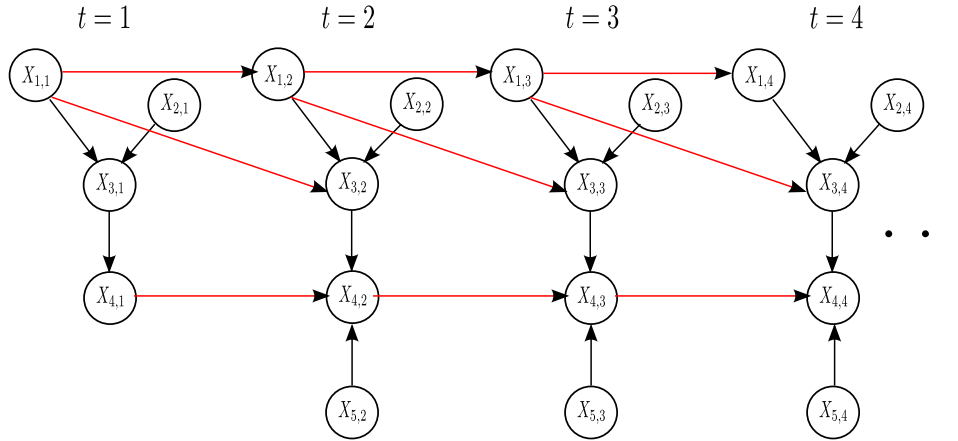


FIGURE 1.4 – Exemple de réseau bayésiens dynamiques

où  $X_{\text{pa}_{n,t}}$  désigne l'ensemble des parents de  $X_{n,t}$  dans le graphe .

En exploitant les hypothèse ci-dessus et les indépendances au sein d'un état et entre les états, on peut utiliser un réseau bayésien pour représenter le processus de manière plus compacte :

**Définition 1.4.1** *Un 2-TBN (Two time-slice temporal bayesian network) est une classe particulière de RBD. C'est un réseau bayésien à deux tranches de temps défini par un couple de RB  $(M_1, M_{\rightarrow})$  (voir figure 1.5).*

$M_1$  a pour but de décrire la loi initiale des variables

$$M_1 : P(\mathbf{X}_1) = \prod_{n=1}^N P(X_{n,1} | X_{\text{pa}_{n,1}}) \quad (1.39)$$

$M_{\rightarrow}$  est un RB décrivant la loi de transition de l'état du processus à l'instant  $t > 1$  conditionnellement à son état à l'instant  $t - 1$  (la loi de  $X_t | X_{t-1}$ )

$$M_{\rightarrow} : P(\mathbf{X}_t | \mathbf{X}_{t-1}) = \prod_{n=1}^N P(X_{n,t} | X_{\text{pa}_{n,t}}), 1 < t \leq T \quad (1.40)$$

où  $\text{pa}_{n,t}$  désigne les couples d'indices des v.a. parentes de  $X_{n,t}$  dans le graphe de  $M_{\rightarrow}$ .

D'après 1.39 la loi jointe énoncée dans 1.38 s'écrit :

$$P((\mathbf{X}_t)_{1 \leq t \leq T}) = P(\mathbf{X}_1) \prod_{t=2}^T \prod_{n=1}^N P(X_{n,t} | X_{\text{pa}_{n,t}}) \quad (1.41)$$

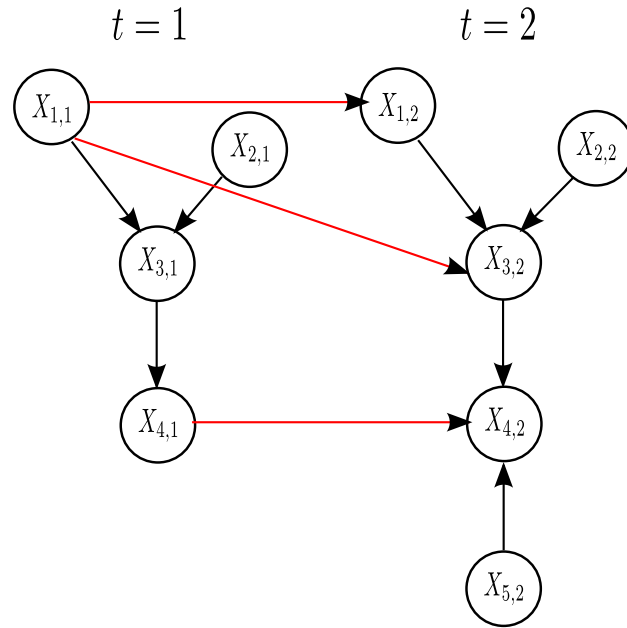


FIGURE 1.5 – Un réseau bayésien dynamique sous sa forme 2-TBN

**Définition 1.4.2** *Un RBD d'horizon  $T$  est un réseau bayésien que l'on obtient en déroulant  $T$  fois le 2-TBN associé. Chaque état  $\mathbf{X}_t$  du RBD est appelé slice au temps  $t$ .*

#### 1.4.1.3 Inférence

Avant d'attaquer le processus d'inférence, il est essentiel de s'assurer que la structure et les tables de probabilités de chaque variable sont connues. Dans la réalité, il n'est toujours pas évident qu'un expert puisse définir la structure et/ou les paramètres du RB. Deux cas sont alors envisagés :

- La structure du réseau est définie par l'expert : les tables de probabilités pourront être définies par apprentissage, à partir de données de retours d'expérience.
- Aucune hypothèse n'est disponible quant à la structure du réseau bayésien : dans ce cas la structure du réseau est construite par apprentissage. Le but de cet apprentissage est de trouver une structure qui, une fois munie des paramètres les plus appropriées, représente au mieux les dépendances entre les grandeurs du système.

Dans un réseau bayésien, les algorithmes d'inférence utilisent la notion de probabilité conditionnelle ainsi que le théorème de Bayes. L'inférence se résume alors à un calcul des probabilités a posteriori : connaissant les états de certaines variables (observation), on détermine les probabilités des états d'autres variables cibles, conditionnellement aux observations. Plusieurs algorithmes d'inférence ont été développés pour les MGPs. Certains utilisent des méthodes d'inférence

exactes, d'autres des méthodes d'inférence approchées.

#### Inférence dans les réseaux bayésiens statiques :

Les méthodes d'inférence exactes utilisent les indépendances conditionnelles dans le réseaux et donnent à chaque inférence des probabilités a posteriori exactes. Les premiers versions de ces méthodes sont basées sur une architecture à passage de messages [Pearl 1986] se limitant aux arbres. Dans cette technique, chaque nœud envoie des messages à ses voisins d'une manière asynchrone jusqu'à l'établissement d'un équilibre

Cette méthode a été, étendue aux graphes quelconques ce qui a aboutit à la méthode de l'arbre de jonction [Lauritzen 1988] qui est largement utilisée. Il y a aussi la méthode d'élimination de variables [Dechter 1999] qui est simple et facile à mettre en œuvre et les méthodes symboliques [Li 1994] permettant d'exprimer littéralement la solution dans les cas les plus simples.

Pour limiter les quantités de calcul, les méthodes d'inférence exacte tendent à traiter les variables de manière locale. Néanmoins, cette simplification est insuffisante pour traiter des réseaux trop complexes. Par ailleurs, [Cooper 1990] montre que l'inférence exacte dans des réseaux quelconques est un problème NP-difficile d'où le recours aux méthodes d'inférence approchée.

Les méthodes approchées font une estimation des probabilités a posteriori. Par conséquent, deux exécutions de l'inférence sur un même réseau pourraient donner des probabilités a posteriori différentes. Parmi les approches les plus utilisées pour réaliser l'inférence approchée on trouve celles basées sur le principe de *Monte Carlo Markov Chain* MCMC [Gil 1995], [Bremaud 2001], la méthode *Loopy belief propagation* [Murphy 1999], les méthodes variationnelles [Jordan 1999], [Lawrence 2000], [Bishop 2002] et la méthode du filtrage particulière [Arulampalam 2002]. Les méthodes d'échantillonnage de Gibbs ou de Metropolis-Hastings [Lauritzen 1996] peuvent aussi être appliquées aux réseaux bayésiens.

La MCMC exploite la topologie du réseau et effectue un échantillonnage de Gibbs sur des sous-ensembles locaux de variables de façon séquentielle et concurrente. Les méthodes dites variationnelles cherchent, quant à elles, à déterminer le maximum de vraisemblance en approximant la probabilité a posteriori. L'approximation peut aussi s'opérer seulement sur un sous ensemble de variables [Draper 1995] ou bien en évaluant les sommes impliquées durant une inférence de type exact [D'Ambrosio 1993]. On peut, de même, limiter le réseau sur lequel se fait l'inférence en ignorant les dépendances les plus faibles [Kjoerulff 1994].

### Inférence dans les réseaux bayésiens dynamiques

Une fois le réseau bayésien dynamique spécifié, plus précisément dans sa représentation 2-TBN, le travail de l'inférence peut commencer. Nous sommes dans l'hypothèse que toutes les variables cibles du RBD sont discrètes. Disposant des observations  $y_{1:t}$ , telles que  $1 \leq t \leq T$  et selon la nature de la requête probabiliste à laquelle on s'intéresse, plusieurs types d'inférences sont possibles, parmi eux :

- pour  $P(X_t|y_{1:t})$  : filtrage, méthode *on-line* qui estime l'état courant du modèle, à partir de toutes les observations jusqu'au temps courant  $t$  ;
- pour  $P(X_{1:t}|y_{1:t})$  : lissage *off-line*, méthode qui estime tous les états passés à partir de toutes les observations jusqu'au temps courant  $t$  ;
- pour  $P(X_{t-\partial t}|y_{1:t})$  lissage, méthode *on-line* qui estime l'état à un certain instant ( $t - \partial t$ ) sur la base de toutes les observations jusqu'au temps courant  $t$  ;
- pour  $P(X_{t+\partial t}|y_{1:t})$  : prédiction, méthode *on-line* qui extrapole la distribution probabiliste pour les tranches de temps futures ;
- pour  $\max_{x_{1:t}}(P(X_{1:t}|y_{1:t}))$  : viterbi, méthode *off-line* qui calcule la séquence la plus probable des états cachés à partir des observations ;

Néanmoins, ces requêtes particulières peuvent se ramener au problème plus général du calcul de la distribution d'un vecteur aléatoire  $(X_{rt})_{1 \leq t \leq T}$  extrait du vecteur  $(X_t)_{1 \leq t \leq T}$ .

### Méthodes d'inférence exactes pour les RBD

La méthode d'inférence la plus simple, mais pas forcément avantageuse dans tous les cas, consiste à transformer le RBD en sa chaîne de Markov équivalente. Cette méthode permet de profiter directement des nombreuses méthodes d'inférences développées pour les chaînes de markov. La complexité spatiale de cette approche peut s'avérer être une contrainte dans la mesure où la propriété de factorisation des modèles graphiques n'est plus utilisée et qu'il faut stocker la loi initiale  $M_1$  et la loi de transition  $M_{\rightarrow}$  pour chaque requête. Si par exemple,  $X$  a  $K$  états possible, le lissage requiert alors  $O(K^2)$  opérations par pas de temps, car l'ensemble des LPC de transition devrait être stockée sur  $T$  tranches. La complexité du lissage sera donc de l'ordre de  $O(K^2T)$ .

La deuxième approche simple, pour faire l'inférence sur un RBD d'horizon temporel  $T$ , consiste à dérouler ce dernier sur  $T$  tranches de façon à obtenir un RB statique. N'importe quel algorithme d'inférence adapté au cas statique pourrait alors être utilisé. Néanmoins, cette technique d'inférence n'exploite pas l'hypothèse markovienne ce qui amplifie la complexité spatiale (surtout dans la cas d'un RBD avec un horizon très grand et un large espace d'état).

Une approche plus "habile" consiste à profiter de la propriété de Markov : le calcul probabiliste dans la tranche  $t$  est possible dès que la distribution jointe



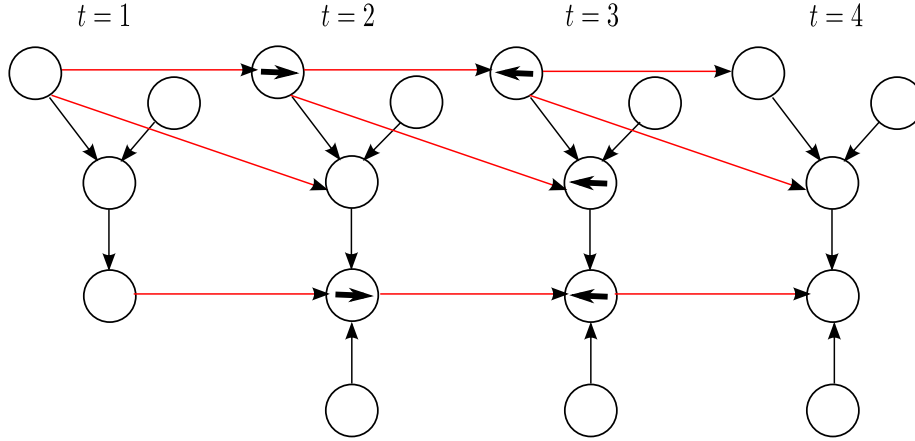


FIGURE 1.6 – Un exemple de réseau bayésien dynamique, où sont représentés l'interface droite ( $\rightarrow$ ) à  $t = 2$  et l'interface gauche ( $\leftarrow$ ) à  $t = 2$ .

des variables des tranches précédentes ou suivantes de  $t$  est connue. L'algorithme de Frontière [Zweig 1996] est fondé sur ce dernier principe. Encore mieux, [Darwiche 2001] démontre que seule l'observation d'une partie des variables appartenant aux tranches précédentes de  $t$  est nécessaire pour assurer l'indépendance entre le passé et le futur du processus. Ces variables constituent ce que l'on appelle l'interface [Murphy 2002] comme illustré dans la figure 1.6.

L'interface est un sous-ensemble de nœuds tel que s'ils étaient supprimés, les informations du passé n'auraient plus aucune influence sur le futur. Ainsi, on peut dire que les nœuds d'interface sont les nœuds destination d'un arc temporel et les parents de ces nœuds.

**Définition 1.4.3** Soit un RBD d'horizon  $T$ .

On appelle interface droite  $X_t^{\text{int} \rightarrow}$ , l'ensemble des nœuds au temps  $t < T$  qui ont au moins un enfant au temps  $t + 1$ .

On appelle interface gauche  $X_t^{\text{int} \leftarrow}$ , l'ensemble des nœuds au temps  $t > 0$  tel que eux ou un de leurs enfants aient un parent au temps  $t - 1$ .

**Proposition 1.4.1** Soit un vecteur aléatoire  $(X_t)_{t \geq 1}$  dont la distribution est représentée par un RBD d'horizon  $T$ . On note respectivement  $X_t^{\text{int} \rightarrow}$  et  $X_t^{\text{int} \leftarrow}$  les interfaces droite et gauche du RBD à tranche  $t$ . Alors pour tout  $t > 0$ ,

$$X_t \perp\!\!\!\perp X_{t+2} \mid X_t^{\text{int} \rightarrow}, \text{ pour tout } t > 0$$

$$X_{t-2} \perp\!\!\!\perp X_t \mid X_t^{\text{int} \leftarrow}, \text{ pour tout } t > 0$$

Preuve : soit  $I$  un nœud de l'interface droite  $X_t^{\text{int}}$ . Supposons que  $I$  est connecté à un nœud  $P$  du passé et à un nœud  $F$  du futur (enfant de  $I$ ). Si  $P$  est parent, on aura alors un graphe tel que  $P \longrightarrow I \longrightarrow F$ . Si  $P$  est un descendant (enfant) on aura alors un graphe tel que  $P \longleftarrow I \longrightarrow F$ .

De toutes façons, on aura toujours  $P \perp\!\!\!\perp F \mid I$  (le fond de la structure en  $V$  n'est pas  $I$ ). Tant que le chemin entre n'importe quelle variable du passé et n'importe quelle variable du futur passe forcément par une variable appartenant à l'interface, le résultat reste vérifié.

La définition suivante généralise la notion d'interface gauche et droite précédemment présentées.

#### Définition 1.4.4

Soit un RBD d'horizon  $T$ .

On appelle  $X_t^{\text{int}}$  l'interface de  $X_t$ , l'ensemble des nœuds d'un RBD d'horizon  $T$  tel que tous les chemins entre n'importe quelle variable du passé ( $X_{t-\partial t}$ ) et n'importe quelle variable du futur ( $X_{t+\partial t}$ ) passent forcément par une variable appartenant à l'interface  $X_t^{\text{int}}$  dans le RBD "moralisé"<sup>4</sup>

Ce résultat permet de réduire la complexité spatiale lors de l'inférence dans un RBD. Il suffit alors de calculer la distribution de l'interface comme le montre la figure 1.7 suivante.

#### Exploitation de l'interface

Soit un RBD d'horizon  $T$ , ayant la même forme que celui présenté dans la figure 1.5.  $(\mathbf{X}_t)_{1 \leq t \leq T} = (X_{1,t}, X_{2,t}, X_{3,t}, X_{4,t}, X_{5,t})$  est l'ensemble des variables évoluant dans le temps sur une séquence de longueur  $T$ .

$(\mathbf{Xr}_t)_{1 \leq t \leq T} = (X_{1,t}, X_{3,t})$  est un vecteur de variables dont les composants sont en fait des variables de  $(\mathbf{X}_t)_{1 \leq t \leq T}$ .

But : Calculer la distribution du vecteur  $(\mathbf{Xr}_t)_{1 \leq t \leq T}$

Prérequis :

1. Le 2-TBN définit par le couple  $(M_1, M_{\rightarrow})$ .
2.  $L_{M_1} = \{p_{1,1}, \dots, p_{N,1}\}$  : LPC du modèle initial  $M_1$ .
3.  $L_{M_{\rightarrow}} = \{p_{1,t}, \dots, p_{N,t}\}$  : LPC du modèle de transition  $M_{\rightarrow}$ .
4. Un algorithme d'inférence adapté aux RBs statiques qu'on appelle **Infer**.

**Infer** permet de calculer la loi de distribution d'un vecteur de variable.

---

4. La moralisation consiste à relier les parents d'un même nœud

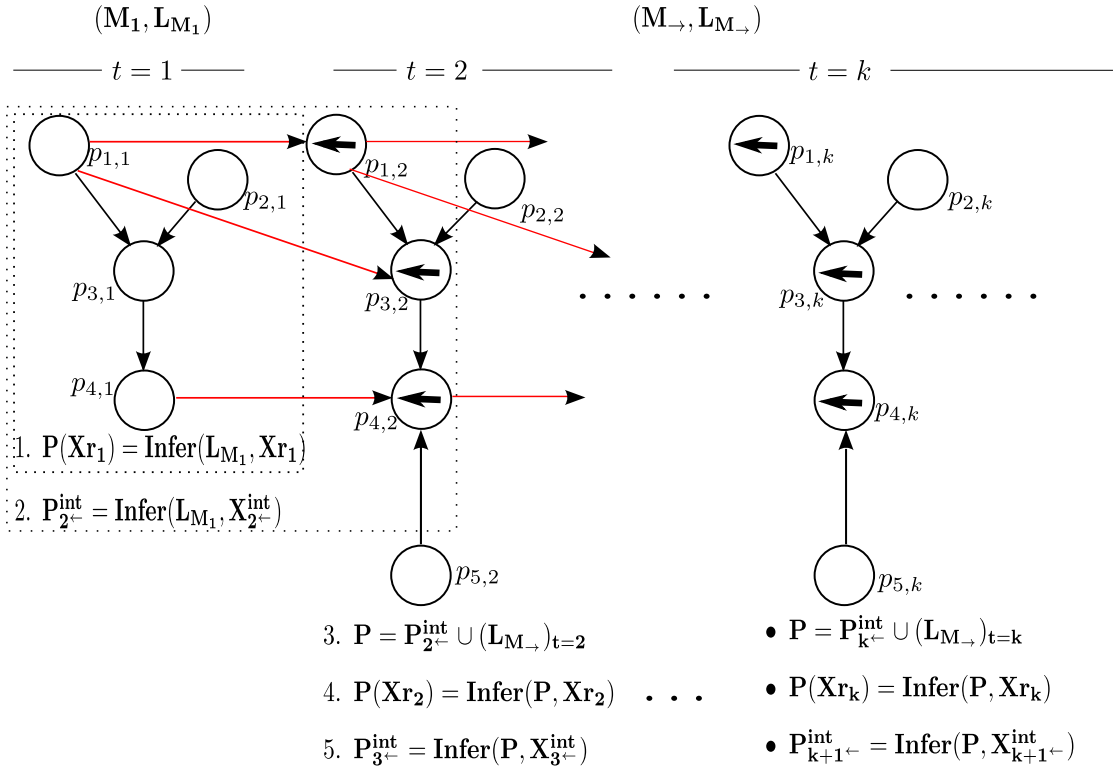


FIGURE 1.7 – Exploitation de l'interface gauche dans le calcul de l'inférence dans un RBD. Les variables avec  $\leftarrow$  composent l'interface gauche

La figure 1.7 décrit une méthode itérative générique permettant de déduire les probabilité d'un vecteur de variables  $(Xr_t)_{1 \leq t \leq T}$  issus du RBD considéré. Le calcul de la distribution de ce vecteur, à chaque tranche de temps, est assuré grâce à la propriété de l'interface gauche ce qui permet de réduire significativement la complexité calculatoire lors de l'inférence.

### Méthodes d'inférences approchées pour les RBD

La structure graphique compliquée d'un RBD rend parfois très complexe, voir impossible, l'inférence exacte. Cette complexité algorithmique est d'autant plus importante que l'horizon temporel, le nombre de variables et d'états sont grands. Pour rendre possible le calcul en pratique de l'inférence dans ce cas de RBD, des méthodes d'inférences approchées sont apparues. On distingue alors deux approches d'algorithmes :

- Les algorithmes approchés déterministes
- Les algorithmes approchés stochastiques

Parmi les algorithmes approchés déterministes les plus connus, il y a l'algorithme BK [Boyen 1999] qui consiste à approcher la distribution jointe de l'interface comme un produit de marginales. Seulement, dans certains cas, cette approxi-

mation peut s'avérer insoluble puisque BK effectue une inférence exacte dans le 2TBN. Cette limitation a encouragé le développement de l'algorithme de la frontière factorisée [?]. Dans [Murphy 2002] il est démontré que ces deux algorithmes sont en fait des cas particuliers du *Loopy Belief propagation* [Murphy 1999].

Les algorithmes approchés stochastiques quant à eux reposent, comme dans le cas statique, sur des techniques d'échantillonnage [Arulampalam 2002]. Elles se déclinent de deux façons : *offline* et *online*. Les méthodes *offline* sont souvent fondées sur un important échantillonnage comme dans l'algorithme *likelihood-weighting* et bien le Monte Carlo Markov Chain (MCMC) [Bremaud 2001].

Les méthodes *online* utilisent le filtrage particulaire [Arulampalam 2002], le filtrage bootstrap [Gordon 1993] ou l'algorithme de condensation [Isard 1996]. On peut trouver plus de détails dans [Arulampalam 2002].

Par rapport aux approches déterministes les algorithmes par approche par échantillonnage sont plus simples à mettre en œuvre et s'adaptent à tous les cas dans la mesure où il est possible de combiner des tables de LPC de différents types et avoir un espace d'état extensible. Toutefois, l'inconvénient majeur de cette approche reste sa lenteur d'exécution, ce qui la rend difficile à appliquer aux modèles graphiques avec large espace d'états.

#### 1.4.2 Modèle graphique de durée

Dans la plupart des applications de sûreté de fonctionnement, le but est de modéliser la dynamique d'un système multi-états afin de prédire l'apparition d'états non conformes, voir d'état de panne. Cette problématique peut être en partie résolue par l'utilisation des modèles graphiques probabilistes qui se sont révélés pertinents pour représenter des systèmes complexes et réaliser des études de fiabilité ( [Langseth 2007] et [Boudali 2005]).

L'inconvénient majeur de cette approche reste la contrainte implicite sur les temps de séjour dans les états du système. Dans leurs utilisation standard les MGP modélisent des temps de séjour distribués géométriquement (ou exponentiellement en temps continu). Comme dans toute modélisation markovienne, la modélisation de la dégradation est souvent biaisée par cette hypothèse restrictive. Dans une analyse de fiabilité, une telle erreur peut avoir de graves conséquences, surtout si l'on veut optimiser les paramètres des politiques de maintenance basées sur la fiabilité.

Citons les travaux de [Bouillaut 2009] qui visent à quantifier l'erreur d'estimation de fiabilité due à une approximation exponentielle lorsque le système suit en réalité d'autres types de distributions de temps de séjour.

Ce problème peut être contourné par l'utilisation des Modèles Graphiques de Durée (MGD) [Donat 2010] qui permettent de prendre en compte n'importe quelle loi de temps de séjour.

Le MGD est un MGP ayant une structure particulière où les temps de séjour sont explicitement représentés par un nœud du réseau. Il se base sur les modèles à variables de durée introduits par [Murphy 2002] afin de construire un modèle discret et fini, où chaque loi de probabilité conditionnelle (LPC) est définie sur un domaine discret et fini. Un tel modèle permet de représenter l'évolution d'un système dynamique au cours du temps. Cette structure prend en compte les différents types d'actions agissant sur les états du système et par conséquent sa loi de durée.

Considérons  $\mathbb{X}$  l'ensemble des états qu'un système peut prendre dans le temps  $\mathbb{X} = \{1, \dots, K\}$ . Parmi lesquels

- l'état de bon fonctionnement,
- les états de dégradation,
- et les états de défaillance.

Lors d'une situation de dégradation, le système continue à fonctionner mais avec une cadence ou un rendu médiocre. On peut par exemple prendre le cas d'un système de production où la dégradation d'un composant de la chaîne de production altère la qualité et ou la quantité des pièces produites. Un autre exemple est celui de la dégradation d'une voie ferrée (usure et fatigue des rails avant la cassure ou bien altération des traverses béton) ce qui astreint les trains à diminuer leur vitesse.

Les descriptions graphique et probabiliste du MGD, sont abordées respectivement dans les parties 1.4.2.1 et 1.4.2.2. Par la suite, nous mettons l'accent sur la méthode d'inférence probabiliste efficace dédiée à cette structure particulière.

#### 1.4.2.1 Définition et structure

La structure du MGD à des fins de modélisation d'une politique de maintenance est représentée par la figure ci-dessus. Ce modèle repose sur trois variables aléatoires  $X_t$ ,  $X_t^D$  et  $A_t$  à valeurs respectivement dans les ensembles  $\mathbb{X}$ ,  $\mathbb{X}^D$  et  $\mathbb{A}$  :

1.  $(X_t)_{1 \leq t \leq T}$  représente l'état du système sur une séquence de longueur  $T$ .
2.  $(X_t^D)_{1 \leq t \leq T}$  représente à chaque instant  $t$ , la durée restante avant un changement d'état du système. Autrement dit,  $X_t^D$  est la variable associée au temps de séjour dans l'état courant. On dit aussi que  $X_t^D$  est la variable de durée à l'instant  $t$ . Elle est discrète et fini car les durées sont supposées être multiples du pas de temps choisi.
3.  $(A_t)_{1 \leq t \leq T}$  représente l'action sélectionnée sur une séquence de temps de

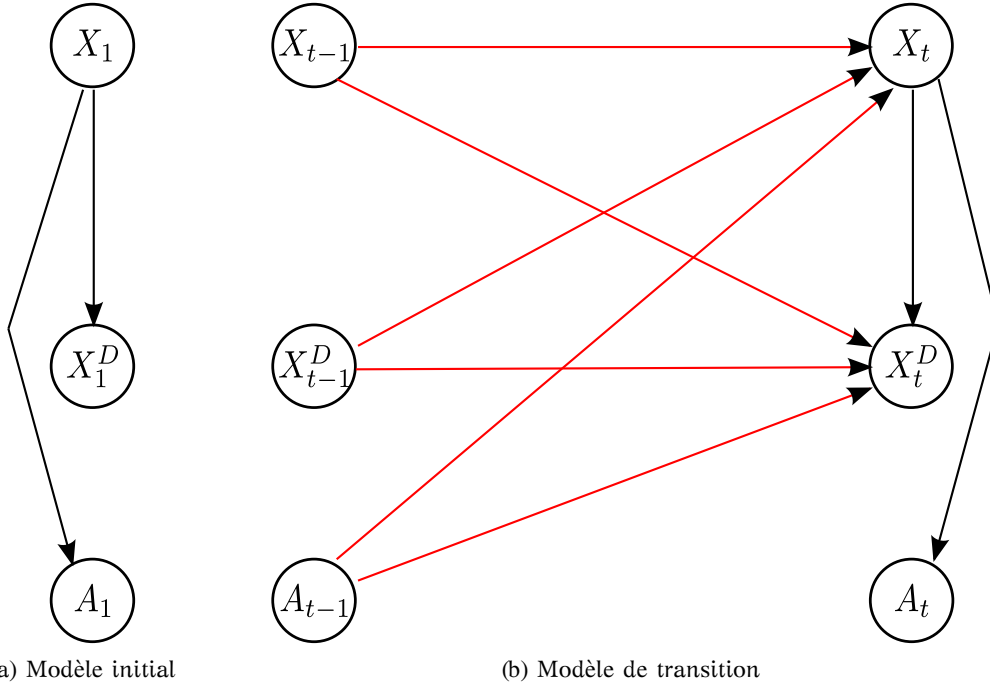


FIGURE 1.8 – Structure du MGD.  $X_t$ ,  $X_t^D$  et  $A_t$  représentent respectivement à l'instant  $t$ , l'état du système, le temps de séjour restant et l'action sélectionnée. Les figures (a) et (b) donnent respectivement les structures du modèle initial et du modèle de transition.

longueur  $T$ . L'action  $A_t$  est sélectionnée conditionnellement à l'état courant  $X_t$ . Ceci permet de spécifier une politique d'action impactant sur l'état du système. Notons l'ensemble des actions possibles  $\mathbb{A}$ . Ce dernier est partitionné entre l'ensemble des actions déclenchant une transition de l'état du système, noté  $A^X$ , et celui des actions laissant l'état du système inchangé, noté  $A^{\bar{X}}$ .

D'après les relations de dépendance engendrées par ce modèle nous pouvons constater que le processus  $(X_t, X_t^D, A_t)_{1 \leq t \leq T}$  est markovien contrairement au processus  $(X_t)_{1 \leq t \leq T}$  qui en fait, semi-markovien (Car  $X_t$  ne dépend plus uniquement de l'état précédent) [Howard 2007].

D'après la définition 1.4.3, les variables formant l'interface gauche et l'interface droite à l'instant  $t$  sont respectivement :

- $X_{t-1}^{\text{int}} = (X_{t-1}, X_{t-1}^D, A_{t-1})$
- $X_t^{\text{int}} = (X_{t+1}, X_{t+1}^D)$ .

Conformément à la proposition 1.4.1 appliquée au MGD, on a les subordinations suivantes :

$$X_t, X_t^D, A_t \perp\!\!\!\perp X_{t+2}, X_{t+2}^D, A_{t+2} \mid X_{t+1}, X_{t+1}^D \quad (1.42)$$

$$X_{t-2}, X_{t-2}^D, A_{t-2} \perp\!\!\!\perp X_t, X_t^D, A_t \mid X_{t-1}, X_{t-1}^D, A_{t-1} \quad (1.43)$$

#### 1.4.2.2 Description probabiliste

Chaque LPC peut se mettre sous la forme d'une matrice stochastique. Pour plus de clarté, nous allons étudier dans ce qui suit les lois de chaque variable du modèle 1.8b.

##### Estimation de l'état du système

La distribution probabiliste de l'état du système sur une séquence de longueur  $T$  est définie par la loi de probabilité initiale et la loi de transition.

##### Distribution initiale

D'abord, il faut spécifier la distribution initiale de l'état du système à  $t = 1$ . La LPC associée à  $X_1$  est définie pour tout  $k \in \mathbb{X}$  par

$$E_1(k) = P(X_1 = k) \quad (1.44)$$

$P(X_1 = k)$  est définie sur une seule dimension ou sous la forme d'un vecteur à  $K$  éléments.

##### Loi de transition

La LPC caractérisant les transitions de l'état  $X_{t-1}$  vers l'état  $X_t$ , lorsque  $t \geq 2$  est maintenant à définir. Trois cas sont à considérer selon le temps de séjour restant et l'action sélectionnée à l'instant précédent :

1. Si la durée restante avant le changement de l'état du système est  $X_{t-1}^D = 1$  (le temps de séjour à l'instant  $t-1$  est écoulé) et l'action sélectionnée n'agit pas sur le système ( $A_{t-1} \in A^{\bar{X}}$ ), on a alors

$$P(X_t = k \mid X_{t-1} = k', X_{t-1}^D = 1, A_{t-1} \in A^{\bar{X}}) = E^{\text{sys}}(k', k) \quad (1.45)$$

où  $E^{\text{sys}}$  est une matrice carrée stochastique de  $|\mathbb{X}|$  lignes et  $|\mathbb{X}|$  colonnes que nous supposons stationnaire.

2. En revanche, tant que la durée restante de l'état courant avant le changement de l'état du système est  $X_{t-1}^D \geq 2$ , et que l'action sélectionnée n'agit pas sur le système ( $A_{t-1} = a' \in A^{\bar{X}}$ ), celui-ci reste dans l'état courant. Dans

ce cas, la loi de transition est déterministe et la matrice de transition est égale à la matrice identité, notée  $I$ . Ainsi, pour tout  $k, k' \in \mathbb{X}$  on a

$$P(X_t = k | X_{t-1} = k', X_{t-1}^D \geq 2, A_{t-1} = a' \in A^{\bar{X}}) = I(k, k') = \begin{cases} 1 & \text{si } k = k', \\ 0 & \text{sinon,} \end{cases} \quad (1.46)$$

3. Si l'action sélectionnée agit sur le système  $A_{t-1} \in A^X$ , une transition est alors déclenchée, quel que soit le temps de séjour restant de l'état courant. La loi de transition définie pour  $k, k' \in \mathbb{X}$  est la suivante :

$$P(X_t = k | X_{t-1} = k', X_{t-1}^D = 1, A_{t-1} = a') = E^{\text{act}}(k', a', k) \quad (1.47)$$

où  $E^{\text{act}}$  est une matrice stochastique de  $|\mathbb{X}| \times |A^X|$  lignes et  $|\mathbb{X}|$  colonnes. La valeur de  $E^{\text{act}}(k', a', k)$  correspond à la probabilité de passer de l'état  $k'$  à l'état  $k$  sachant que l'action déclenchée au pas précédent était  $a'$ .

Ainsi, d'après les équations 1.45, 1.46 et 1.47 la loi de transition globale de l'état du système est donnée pour tout  $d' \in \mathbb{N}$ , tout  $k, k' \in \mathbb{X}$  et tout  $a' \in A$  par :

$$P(X_t = k | X_{t-1} = k', X_{t-1}^D = d', A_{t-1} = a') = E_t(k, k', d', a') = \begin{cases} E^{\text{sys}}(k', k) & \text{si } d' = 1 \text{ et } a' \in A^{\bar{X}}, \\ I(k, k') & \text{si } d' \geq 2 \text{ et } a' \in A^{\bar{X}}, \\ E^{\text{act}}(k', a', k) & \text{si } a' \in A^X \end{cases} \quad (1.48)$$

### Estimation du temps de séjour restant

Il s'agit à présent de définir la LPC associée aux temps de séjour restant dans les différents états du système. On suppose que  $X_t^D$  est à valeurs dans  $\mathbb{X}^D = \{1, \dots, d, \dots, D\}$ . Notons qu'il n'est pas naturel de définir une v.a représentant une durée sur un ensemble borné. En effet, il aurait été plus convenable de définir  $X_t^D$  sur  $\mathbb{N}$  mais dès lors, le modèle ne serait plus fini. En pratique, il suffit de choisir une borne  $D$  suffisamment grande comparée à la dynamique du système étudié.

### Distribution initiale

Commençons par définir la distribution initiale des temps de séjour de chacun des états en posant pour tout  $k \in \mathbb{X}$  et tout  $d \in \mathbb{X}^D$ ,

$$P(X_1^D = d | X_1 = k) = F_1(k, d) \quad (1.49)$$

où  $F_1$  est une matrice de  $|\mathbb{X}|$  lignes et de  $D$  colonnes.  $F_1(k, d)$  donne la probabilité que le système reste  $d$  unités de temps dans l'état  $k$ .



*Loi de transition*

La LPC de  $X_t^D$  pour  $t \geq 2$  va permettre de mettre à jour les temps de séjour qui deviennent ainsi la durée restante avant la prochaine transition.

Deux cas sont à étudier :

1. Lorsque le temps restant dans l'état précédent  $k'$  est écoulé, i.e.  $X_{t-1}^D = d' = 1$ , ou bien lorsque l'action sélectionnée agit sur le système ( $A_{t-1} \in A^X$ ), une transition ayant pour conséquence un changement d'état est déclenchée.

La distribution du temps de séjour dans le nouvel état est alors donnée pour tous  $k', k \in \mathbb{X}$ , et tout  $d \in \mathbb{X}^D$ , par :

$$P(X_t^D = d \mid X_{t-1}^D = 1, X_t = k' \text{ ou } A_{t-1} \in A^X, X_t = k) = F^{\text{sys}}(k', k, d) \quad (1.50)$$

où  $F^{\text{sys}}$  est une matrice de  $|\mathbb{X}|^2$  lignes et  $D$  colonnes. Tout comme  $F_1$ , chaque ligne de la matrice  $F^{\text{sys}}$  échantillonne à une loi de temps de séjour dépendant de l'état courant, et également de l'état précédent.

2. Dans le cas où il reste du temps avant un changement d'état, et que l'action sélectionnée n'agit pas sur le système i.e.  $X_{t-1}^D \geq 2$ ,  $A_{t-1} \in A^{\bar{X}}$ , le système reste dans l'état précédent  $k'$  et la durée restante dans ce même état est décrétementée de façon déterministe d'une unité. D'où, pour tout  $d, d' \in \mathbb{X}^D$ ,

$$P(X_t^D = d \mid X_{t-1}^D = d' \geq 2, A_{t-1} \in A^{\bar{X}}) = C(d', d) \quad (1.51)$$

où  $C$  est une matrice carrée de  $D$  lignes vérifiant pour tout  $d' \in \mathbb{X}^D$ ,

$$C(d', d) = \begin{cases} 1 & \text{si } d = d' - 1, \\ 0 & \text{sinon} \end{cases} \quad (1.52)$$

Par conséquent, d'après les dernières équations, la LPC de  $X_t^D$ ,  $t \geq 2$  est définie pour tout  $d, d' \in [1 \dots D]$ , tout  $k, k' \in \mathbb{X}$  et tout  $a' \in \mathbb{A}$  par :

$$P(X_t^D = d \mid X_{t-1} = k', X_{t-1}^D = d', A_{t-1} = a', X_t = k) =$$

$$F_t(k', d', a', k, d) = \begin{cases} F^{\text{sys}}(k', k, d) & \text{si } d' = 1 \text{ ou } A_{t-1} \in A^X, \\ C(d', d) & \text{si } d' \geq 2 \text{ et } A_{t-1} \in A^{\bar{X}} \end{cases} \quad (1.53)$$

**Politique d'action**

La LPC associée à la variable  $A_t$  permet de décrire la politique d'action agissant sur le système. L'ensemble des actions possibles est noté  $\mathbb{A}$ . Ce dernier est partitionné entre l'ensemble des actions déclenchant une transition de l'état du

système et celui des actions laissant l'état du système inchangé. Cette LPC vérifie pour tout  $k \in \mathbb{X}$  et tout  $a \in \mathbb{A}$ ,

$$P(A_t = a \mid X_t = k) = G_t(k, a). \quad (1.54)$$

La valeur  $G_t(k, a)$  donne la probabilité de sélectionner l'action  $a$  sachant que le système se trouve dans l'état  $k$ . Cette LPC peut se mettre sous la forme d'une matrice stochastique de  $|\mathbb{X}|$  lignes et  $|\mathbb{A}|$  colonnes. Lorsque la politique d'action ne dépend pas de  $t$ , ( $G_t = G$ ), nous parlons de politique homogène. Dans la plupart des politiques de maintenance  $G$  est déterministe et les actions déclenchées sont certaines selon l'état du système (chaque ligne de  $G$  contient une valeur à 1 et les autres valeurs à 0)

### 1.4.2.3 Inférence probabiliste

Comme nous l'avons évoqué précédemment (1.4.2.1), dans le cadre des modèles de durée, l'étude d'un système dynamique se ramène à l'estimation des lois décrivant ses transitions et ses lois de durée, et ce, à n'importe quel instant.

Par ailleurs, la relation d'indépendance conditionnelle citée précédemment (1.42) implique la nécessité de connaître la distribution  $\Psi_t = P(X_t, X_t^D, A_t)$  avant tout calcul d'inférence à l'instant  $t+1$ . Le théorème de factorisation dans le MGD associé à l'équation (1.43) conduit à la définition récursive suivante pour l'interface gauche :

$$\Psi_t = \begin{cases} P(X_1)(X_1^D|X_1)P(A_1|X_1) & t = 1 \\ \sum_{\substack{x_{t-1}, x_{t-1}^D \\ A_{t-1}}} [\Psi_{t-1}P(X_t|X_{t-1}, X_{t-1}^D, A_{t-1}) & t \geq 2 \\ P(X_t^D|X_{t-1}, X_{t-1}^D, A_{t-1}, X_t)]P(A_t|X_t) & \end{cases} \quad (1.55)$$

Le problème soulevé ici a pu être résolu par un algorithme d'inférence générique adapté aux 2TBN tel que l'algorithme de l'interface décrit par [Murphy 2002]. Cependant, lorsque la taille de la séquence  $T$  devient grande ou encore lorsque la borne des temps de séjour  $D$  augmente, les méthodes génériques deviennent très rapidement inutilisables. Une méthode itérative *ad hoc* a été alors développée par [Donat 2010]. Cette méthode, détaillée dans l'algorithme 1 ci-après, est capable de calculer efficacement les distributions introduites dans 1.48, 1.53 et 1.54 en exploitant les parties déterministes de chacune des LPC impliquées. L'objectif de l'étape 2 de l'algorithme est d'affiner le calcul de l'interface gauche  $\Psi_t$ . La démonstration de l'équation 1.56 de cette même étape de l'algorithme est fondée sur l'application de l'algorithme d'élimination [Dechter 1999] sur le MGD. L'idée clé est d'exploiter la propriété de la factorisation de la distribution jointe 1.55

---

**Algorithme 1** : Algorithme calculant la distribution de probabilité d'un ensemble de variable à chaque instant  $t$  du MGD avec méthode *ad hoc*

---

**Données** : L'ensemble des LPC associées au MGD caractérisant le processus  $(X_t, X_t^D, A_t)_{t \geq 1}$ , où les variables  $X_t$ ,  $X_t^D$  et  $A_t$  sont respectivement à valeur dans  $\mathbb{X}$ ,  $\mathbb{S}$  et  $\mathbb{A}$ .

**Résultat** : Calculer  $P(\Psi_t(k, d, a))$

1. Calculer  $P(\Psi_1(k, d, a)) = \sum_{\substack{k \in \mathbb{X}, d \in \mathbb{X}^D \\ a \in \mathbb{A}}} (E_1(k), F_1(k, d), G_1(k, a))$
2. Simplifier l'expression de  $(\Psi_t(k, d, a))_{t \geq 2}$  en exploitant les parties déterministes de chacune des LPC impliquées pour  $t \geq 2$ .

$$\begin{aligned} \Psi_t(k, d, a) &= \sum_{\substack{k' \in \mathbb{X}, d' \in \mathbb{X}^D \\ a' \in \mathbb{A}}} \Psi_{t-1}(k', d', a') E_t(k, k', d', a') F_t(d, d', k', a', k) G_t(k, a) \\ &= (\Psi_t^{\text{sys}}(k, d) + \Psi_t^{\text{det}}(k, d) + \Psi_t^{\text{act}}(k, d)) G_t(k, a) \end{aligned} \quad (1.56)$$

tel que

$$\Psi_t^{\text{sys}}(k, d) = \begin{cases} \sum_{k' \in \mathbb{X}} E^{\text{sys}}(k', k) F^{\text{sys}}(k', k, d) \sum_{a' \in \mathbb{A}^{\bar{X}}} \Psi_{t-1}(k', 1, a') & \text{si } 1 \in \mathbb{X}^D \\ 0 & \text{sinon} \end{cases}$$

$$\Psi_t^{\text{det}}(k, d) = \begin{cases} \sum_{a' \in \mathbb{A}^{\bar{X}}} \Psi_{t-1}(k, d+1, a') & \text{si } k \in \mathbb{X} \text{ et } d \in \mathbb{X}^D \\ 0 & \text{sinon} \end{cases}$$

$$\Psi_t^{\text{act}}(k, d) = \sum_{k' \in \mathbb{X}} F^{\text{sys}}(k', k, d) \sum_{a' \in \mathbb{A}^X} E^{\text{act}}(k', a', k) \sum_{d' \in \mathbb{X}^D} \Psi_{t-1}(k', d', a')$$

3. **pour**  $\{t = 2, \dots, T\}$  **faire**

(a) Calculer  $\Psi_t(k, d, a)$

(b) Dédire  $P(\Psi_t(k, d, a)) = \sum_{\substack{k \in \mathbb{X}, d \in \mathbb{X}^D \\ a \in \mathbb{A}}} (\Psi_t(k, d, a))$

**fin**

---

et les propriétés de l'interface 1.43 pour trouver un caractère récursif dans le temps. Une démonstration est détaillée dans la thèse [Donat 2009]. Cette simplification d'écriture de l'équation permet de réduire significativement la complexité algorithmique en profitant du caractère défini par morceaux et des composantes déterministes des LPC  $E_t$  et  $F_t$ .

## 1.5 Conclusion

Le chapitre 1 présente une étude de l'état d'art des méthodes de modélisation de la dégradation. L'accent a été mis sur les approches de modélisation par modèles graphiques probabilistes et notamment le modèle graphiques de durée (MGD). La structure graphique particulière des MGD permet de dépasser les limitations sous-jacentes à l'utilisation des chaînes de Markov. En effet, les MGD permettent de prendre en compte n'importe quelle distribution de temps de séjour alors qu'une chaîne de Markov impose par construction que les durées dans chaque état du système suivent des lois exponentielles.

# Construction d'une mesure de performance pour l'évaluation des stratégies de maintenance

---

## Sommaire

<b>2.1 Introduction</b>	<b>39</b>
<b>2.2 Enjeux de la fonction Maintenance</b>	<b>40</b>
<b>2.3 Définition de la maintenance</b>	<b>40</b>
<b>2.4 Théorie de la maintenance</b>	<b>43</b>
<b>2.5 Modélisation de la maintenance</b>	<b>49</b>
<b>2.6 Évaluation des politiques de la maintenance</b>	<b>53</b>
<b>2.7 Conclusion</b>	<b>61</b>

---

## 2.1 Introduction

Dans le contexte économique et industriel actuel, les entreprises sont en quête perpétuelle de compétitivité, et de croissance durable ; la réduction de leurs coûts et la maximalisation de la disponibilité de leurs produits sont leurs principaux objets.

Les exigences de durabilité, fiabilité et de sûreté de fonctionnement des systèmes de plus en plus complexes font que les problématiques de maintenance soient une des clés du progrès de l'industrie et des systèmes industriels.

Ces enjeux auxquels il convient d'ajouter les objectifs de sécurité font du secteur de la maintenance une des clés de l'optimisation des systèmes technologiques et industriels.

La première partie de ce chapitre décrit les enjeux industriels et le rôle stratégique de la maintenance. Nous dressons ensuite, un inventaire des différentes politiques de maintenance existantes.

Enfin, nous proposons un modèle d'utilité pour l'évaluation des stratégies de maintenance en les composant à plusieurs autres critères. Cette modèle permettra de quantifier la pertinence d'une politique de maintenance donnée. Le choix de la fonction d'utilité sera guider par sa capacité à prendre en compte simultanément les contraintes économiques et les exigences de sûreté de fonctionnement.

## 2.2 Enjeux de la fonction Maintenance

Pour demeurer compétitive, une entreprise doit proposer un produit ou un service toujours meilleur et au coût le plus bas. Pour minimiser ce coût, il est indispensable de garantir un état de fonctionnement minimale et autant le temps des interruptions de service rendus nécessaire par les opération de remise à niveau.

Cette double contrainte est en partie satisfaite par l'automatisation des inspections calendaires et de la surveillance continue des systèmes mais celles-ci ne peuvent empêcher les ralentissements et les arrêts dues à la dégradation ou la défaillance des systèmes. Ainsi, pour assurer un meilleure service avec de faibles charges et un rendement de meilleure qualité et en continu, l'entreprise doit définir une politique de maintenance afin de répondre aux besoins de sa clientèle et garantir sa profitabilité.

Le maintien en condition opérationnelle des équipements productifs de l'entreprise est ainsi un élément très important pour la bonne marche de l'activité et la prise en charge des risques de défaillance pouvant survenir à tout moment.

Dans ce sens plusieurs entreprises ont engagé une démarche d'Optimisation de la Maintenance basée sur la Fiabilité (OMF) [Zwingelstein 1996]. Cette dernière permet de définir une politique de maintenance selon plusieurs critères comme la disponibilité, la sûreté et les coûts et tout en prenant en considération les conséquences d'une telle stratégie sur l'occurrence des défaillances et le fonctionnement du système. Évidemment la question se pose : Comment reconnaître une bonne stratégie de maintenance ? Comment l'évaluer ?

## 2.3 Définition de la maintenance

L'Association Française de Normalisation (AFNOR) définit la maintenance comme : "L'ensemble de toutes les actions techniques, administratives et de management durant le cycle de vie d'un bien, destinées à le maintenir ou à le rétablir dans un état dans lequel il peut accomplir la fonction requise." En d'autres termes, pour un système industriel assujetti à des dysfonctionnements, les actions de maintenance visent à limiter, voir éviter, les indisponibilités du

système. Ces indisponibilités affectent la sûreté, les coûts et la qualité des services fournis par le système. Par conséquent, la maintenance quand elle est optimale, contribue fortement à limiter les indisponibilités, maîtriser les coûts et la qualité des services et assurer une sécurité maximale des personnes et de l'environnement.

La maintenance s'intègre dans le concept global de la sûreté de fonctionnement, qui regroupe 4 disciplines :

- La Fiabilité (AFNOR X-06-501) : Aptitude d'un dispositif à accomplir une fonction requise dans des conditions d'utilisation données à un instant donné.
- La Disponibilité (AFNOR X-06-010) : Aptitude d'un dispositif à accomplir une fonction requise dans des conditions d'utilisation données pendant une période donnée.
- La Maintenabilité (AFNOR X-06-010) : Aptitude d'un dispositif à être maintenu ou rétabli dans un état dans lequel il peut accomplir une fonction requise lorsque la maintenance est accomplie dans des conditions d'utilisation données avec des moyens et procédures prescrits.
- La Sécurité (AFNOR X-06-010) : Aptitude d'un dispositif à éviter de faire apparaître des événements critiques ou catastrophiques.

La notion de sécurité est capitale dans le nucléaire, le domaine des systèmes de transport, dans les machines outils ou dans les procédés continus comme la pétrochimie.

### 2.3.1 Fiabilité

La fiabilité est l'aptitude d'un système  $S$  à accomplir une fonction requise, dans des conditions données et pendant un intervalle de temps donné. Notons  $R(T)$  la probabilité que  $S$  accomplisse ces fonctions pendant l'intervalle de temps  $[0, T]$  sans subir de défaillance, sachant qu'il n'était pas en panne à l'instant 0.

$$R(T) = P(S(t) = \text{non défaillant}) \quad \forall t \in [0, T] \quad (2.1)$$

$S(t)$  est l'état de  $S$  à l'instant  $t$ . La fiabilité peut être donc caractérisée par la MTBF qui désigne le temps moyen entre défaillances consécutives.

$$MTBF = \frac{\text{Somme des Temps de Bon Fonctionnement}}{\text{nombre de défaillances}} \quad (2.2)$$

La Somme des Temps de Bon Fonctionnement inclut les temps d'arrêt hors défaillance et les temps de micro arrêts.

### 2.3.2 Maintenabilité

Dans les conditions d'utilisation données pour lesquelles il a été conçu, la maintenabilité est l'aptitude d'un bien à être maintenu ou rétabli dans un état dans lequel il peut accomplir une fonction requise, lorsque la maintenance est accomplie dans des conditions données, avec des procédures et des moyens prescrits (NF X60-010).

Pour améliorer la maintenabilité, il est nécessaire de faciliter le diagnostic des pannes et de diminuer les temps d'immobilisation. Dans cette optique, les principaux indicateurs de la maintenabilité sont :

- La moyenne des temps de réparation notée MTTR (*Mean Time To Repair*).

$$MTTR = \frac{\text{Somme des temps de réparation}}{\text{Nombre de réparations}} \quad (2.3)$$

- En considérant que le taux de réparation est constant dans le temps, il peut être écrit en fonction du MTTR comme suit :

$$\text{taux de réparation} = \frac{1}{MTTR} \quad (2.4)$$

### 2.3.3 Disponibilité

La disponibilité d'un système S à un instant t, notée D(t), est l'aptitude de ce dernier à être en état d'accomplir les fonctions qui lui sont assignées, dans les conditions données, à l'instant t.

$$D(t') = P(S(t) \text{ est non défaillant}) \quad (2.5)$$

Le taux de disponibilité peut s'exprimer en fonction de la fiabilité et de la maintenabilité de la manière suivante :

$$\text{taux de disponibilité} = \frac{MTBF}{MTBF + MTTR} \quad (2.6)$$

### 2.3.4 Durabilité

La durabilité est l'aptitude d'un système S à accomplir une fonction requise dans des conditions données d'utilisation et de maintenance, jusqu'à ce qu'un état limite soit atteint.

La durabilité minimum d'un équipement est en général imposée par le maître d'œuvre.



## 2.4 Théorie de la maintenance

Les stratégies de maintenance peuvent être classées en deux grandes catégories : la maintenance corrective (MC) et la maintenance préventive (MP).

La maintenance corrective intervient suite à la défaillance du système alors

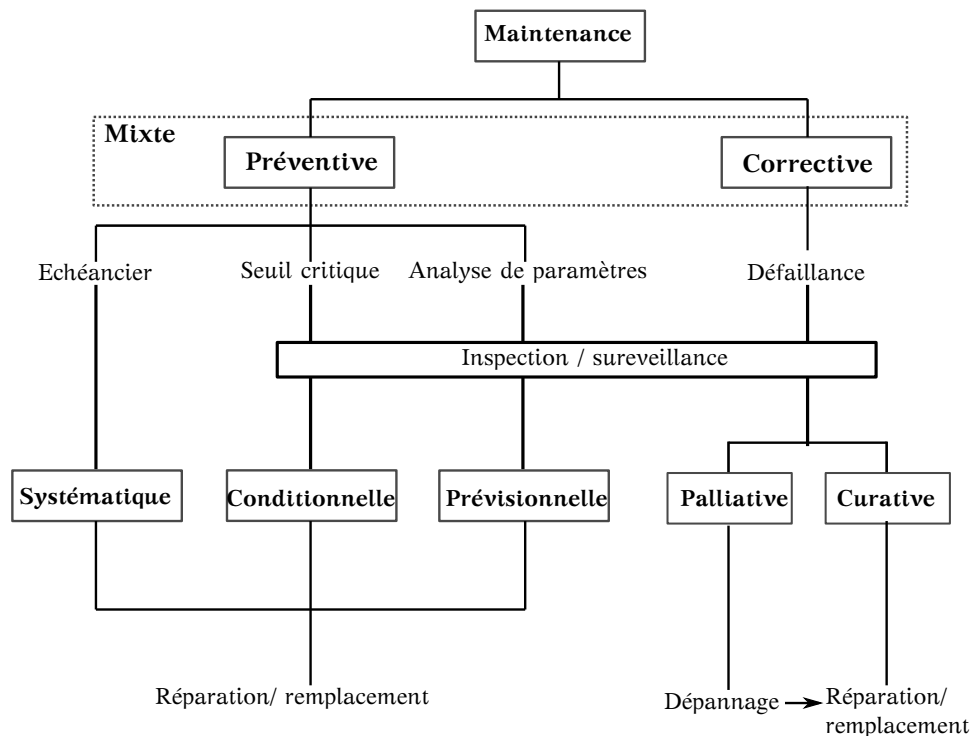


FIGURE 2.1 – Les types de maintenance

que la maintenance préventive est réalisée lorsque le système est encore en fonctionnement. Le recours à l'une ou à l'autre de ces stratégies diffère suivant le type du système considéré mais aussi sa politique d'exploitation, les coûts de réparation, de surveillance et la disponibilité de ces derniers.

Dans la figure 2.1, nous illustrons les différentes stratégies de maintenance.

### 2.4.1 Maintenance corrective

La maintenance corrective, effectuée après la détection d'une panne, est destinée à remettre une entité dans un état lui permettant d'accomplir une fonction requise [Villmeur 1988].

Cette maintenance est utilisée lorsque l'indisponibilité du matériel n'a pas de conséquences majeures sur le rendement du système ou quand les contraintes de sécurité sont faibles [Terrier 2002]. La maintenance corrective peut être utilisée, soit seule, en tant que méthode, ou bien en complément d'une maintenance préventive pour s'appliquer aux défaillances résiduelles. Elle se subdivise en :

- Maintenance curative : la principale étape de la maintenance corrective, elle désigne l'élimination d'une altération dans le fonctionnement d'un équipement (réparation, modification ou remise en état à caractère permanent).
- Maintenance palliative : c'est un dépannage permettant à un équipement de fonctionner, sur le court terme, en attendant une intervention curative pour le remplacement ou la réparation définitive de ce bien (remise en état à caractère provisoire).
- Groupement correctif des actions de maintenance [Sheu 1997] : cette politique est motivée par des économies d'échelle dues à la réparation simultanée de composants identiques. Elles consistent à attendre la défaillance de plusieurs composants d'un système avec des redondances avant de procéder à leur réparation corrective. De cette manière, les matériels en panne sont réparés en même temps et à un moment opportun, pour profiter d'une économie d'échelle. En contrepartie le risque de défaillance du système augmente avec le nombre de matériels défaillants.

### 2.4.2 Maintenance préventive

Une grande partie des travaux scientifiques publiés dans le domaine de la maintenance s'intéressent aux méthodes préventives, qu'elles soient systématiques ou conditionnelles. Les politiques préventives s'avèrent, en pratique, souvent bien plus avantageuses qu'une politique purement corrective entraînant de l'indisponibilité. Toutefois, la Maintenance Préventive (MP) n'est réellement efficace que si elle est correctement optimisée. Citons par exemple les travaux de [Bar-Noy 2004], [Grigoriev 2006] et [Courtois 2006] traitant de l'optimisation de politiques systématiques périodiques. Par ailleurs, les progrès scientifiques sur les capteurs et le diagnostic en général permettent de surveiller les systèmes industriels plus souvent et plus précisément. Ceci explique le nombre croissant d'études dédiées aux politiques de maintenance conditionnelles. En particulier, les travaux de [Grall 2002], [Marseguerra 2002] et [Castanier 2005] s'intéressent à la représentation de telles politiques et à leur optimisation. Bien que performantes d'un point de vue calculatoire, les études précédentes se reposent la plupart du temps sur des hypothèses relativement restrictives quant à la description du système (par exemple sur les lois de dégradation et la représentation du contexte).

La maintenance préventive est effectuée dans l'intention de réduire la probabilité

de défaillance d'un système ou la dégradation d'un service rendu. L'intérêt d'une telle maintenance est principalement d'éviter les périodes de dysfonctionnement avant la panne, ainsi que les pertes éventuelles provoquées par une panne intempestive. Elle permet ainsi de diminuer les travaux urgents, de faciliter la gestion de la maintenance, de favoriser la planification des travaux et d'augmenter la sécurité.

La maintenance préventive se base sur la connaissance des composants, sur leur mode de dégradation et sur les exigences économiques. En effet, elle vise à réduire le coût des pannes et de maintenance, et ce, en se fondant sur le constat que la plupart des réparations et des immobilisations coûteuses auront pu être réduites ou évitées grâce à un entretien régulier et préventif.

La maintenance préventive convient bien à des systèmes clés ou critiques, pour lesquels la panne est prévisible. Nous présentons, dans ce qui suit les différentes approches de la maintenance préventive.

#### 2.4.2.1 Maintenance systématique

La maintenance préventive systématique (MPS) est la maintenance préventive effectuée à des intervalles de temps préétablis (un échéancier) ou selon un nombre défini d'unités d'usage. L'unité d'usage caractérise l'exploitation du système à maintenir. Elle peut être par exemple le nombre de km parcouru par un véhicule ou bien le nombre de pièces fabriquées par une machine.

Lorsque les interventions planifiées sont périodiques, on parle alors de MP Périodique (MPSP/MPP), sinon il s'agit de MP Non Périodique (MPSNP/MPNP). Ce type de maintenance concerne plutôt des équipements dont une défaillance met en cause la sécurité du matériel et des personnes. Il peut concerner aussi des équipements à coût de défaillance élevé, des équipements dont l'arrêt (ou le redémarrage) est long ou bien des équipements soumis à des obligations réglementaires. Selon [Terrier 2002], la mise en place de cette maintenance nécessite :

- Une étude préalable pour déterminer un coût probable,
- Un choix de fréquences fixes d'intervention (en rapport avec la MTBF),
- Une planification des tâches et des mesures de sécurité,
- Une exploitation des résultats : pour l'historique et le réajustement des fréquences d'interventions.

Tous ses paramètres font que cette stratégie de maintenance est difficile à optimiser notamment à cause de sa sensibilité par rapport à la dispersion des instants de défaillance autour de la MTBF (moyenne de temps entre deux défaillances). Toutefois, optimisée, la MPS reste facile à gérer à cause des dates d'intervention fixes; et globalement moins onéreuse qu'une maintenance corrective. En effet, d'une part, la MPS permet d'éviter les détériorations importantes et de diminuer

les risques d'avaries imprévues. D'autre part, elle se justifie par l'historique des pannes (fréquence et importance des défaillances) et par le coût de remplacement et d'interventions systématiques globalement moins important que le coût de remplacement non systématique qui est assorti d'un risque de panne de gravité plus importante.

#### **2.4.2.2 Maintenance conditionnelle**

La maintenance préventive conditionnelle (MPC) est subordonnée au franchissement d'un seuil prédéterminé caractéristiques de la dégradation ou de la baisse de performance d'un système.

Les paramètres significatifs de la dégradation peuvent être des mesures de caractéristiques physiques comme l'épaisseur d'un matériau, la profondeur d'une fissure, le degré d'érosion, la température, pression .... Le franchissement du seuil peut être mis en évidence par différents types de capteurs associés à des procédures de détection plus ou moins simples.

Le déclenchement des interventions repose alors sur la franchissement de seuils critiques (appelés aussi seuils de décision), des actions de maintenance (réparation, de surveillance, de remplacement ...).

La MPC permet de mieux tenir compte des conditions d'utilisation d'un équipement que la maintenance systématique traditionnelle. En effet, d'une part, elle vise à éviter les interventions inutiles induites parfois par la MPS et d'autre part elle accroît la sécurité en suivant l'évolution des débuts d'anomalies dans le temps, afin d'intervenir dans les meilleures conditions. Toutefois, cette stratégie reste généralement onéreuse en raison du coût occasionné par l'exploitation des technologies de pointes (capteurs, analyseurs, instrumentation, logiciels...) lui permettant d'analyser en temps réel l'évolution de l'état d'une entité.

#### **2.4.2.3 Maintenance prévisionnelle**

La maintenance préventive prévisionnelle est subordonnée à la surveillance de l'évolution de paramètres significatifs de dégradation de l'entité, permettant de retarder et de planifier les interventions. Elle se base sur l'estimation du temps de fonctionnement correct de l'entité jusqu'à l'occurrence d'un événement redouté.

Elle peut être considérée comme un type particulier de maintenance conditionnelle comme le stipule la norme AFNOR X-60-319, énoncée comme suit :

“la maintenance prévisionnelle est une maintenance conditionnelle exécutée en

suivant les prévisions extrapolées de l'analyse et de l'évaluation de paramètres significatifs de la dégradation du bien“

Le principe de cette stratégie de maintenance est de projeter l'état actuel de l'entité dans le futur, afin d'estimer le temps de fonctionnement avant la défaillance, et donc de mieux dimensionner les dates d'interventions.

Par conséquent, la maintenance prévisionnelle fournit une planification précise et dynamique des interventions, entraînant ainsi une réduction des pannes et des périodes d'immobilisation. Ce type de maintenance peut s'appliquer à tout types d'équipement dans la mesure où des signes préconiseurs des avaries sont lesquelles et que des modèles(généralement stochastiques)dégradent sont disponibles.

#### 2.4.2.4 Groupement préventif de maintenance

Le groupement des tâches de maintenance [Wildeman 1996], préventif et planifié s'applique aux systèmes multi-composants. Il prévoit d'effectuer plusieurs tâches de maintenance préventive à un même instant, faisant l'hypothèse que la combinaison de ces interventions de maintenance coûte moins cher que si chaque composant est maintenu séparément [Bouvard 2011], [Do Van 2012]. Le but est de réduire les coûts de maintenance fixe et les temps logistiques d'arrêt du système subissant la maintenance et le temps de déplacement de l'équipe de maintenance.

Cette approche permet alors d'améliorer la gestion des ressources comme les pièces de rechange, les équipes d'intervention et les outils nécessaires, tout en maîtrisant l'indisponibilité programmée du système. Le groupement préventif peut se faire également de manière opportuniste [Nachlas 2005], [Laggoune 2009].

#### 2.4.3 Maintenance mixte

La maintenance mixte est appliquée aux systèmes industriels et combine des actions de maintenance préventives et correctives. Excepté dans les domaines sécuritaires, la maintenance préventive, seule, réduit la probabilité de défaillance mais elle demeure toujours un certain nombre de défaillances qui doit être traité par des actions de maintenance corrective.

Il est alors nécessaire d'agrèer des stratégies de maintenance corrective et de maintenance préventive.

#### 2.4.4 Maintenance opportuniste

La maintenance opportuniste concerne les système multi-composants et les travaux s'y rapportant sont encore récents. Elle a pour objectif de tenir compte des interactions entre les composants d'un même système.

Un composant peut subir une action de maintenance préventive lorsque l'on doit effectuer une action de maintenance, préventive ou corrective, sur un autre composant du système. Les interventions sur le système sont alors limitées et la durée d'indisponibilité pour maintenance est réduite, grâce à la combinaison des interventions correctives et préventives.

Comme pour le groupement préventif des tâches de maintenance, les coûts fixes de maintenance peuvent être réduits. Cependant, étant donné que la date de maintenance n'est pas prévue, les travaux de préparation et en particulier le dimensionnement des ressources est plus difficile. La maintenance opportuniste est un aspect intéressant de réduction des coûts de maintenance et la modélisation des dépendances entre matériels permet des travaux d'optimisation des politiques de maintenance, [Wang 2002, Castanier 2003, Chang 2007]. Elle est particulièrement adaptée lorsque les lois de dégradation de plusieurs composants s'avèrent couplées (la répartition d'un composant accélère ou ralentit la dégradation d'un autre). En maintenance ferroviaire par exemple, des actions de maintenance opportuniste peuvent être effectuées dans une zone voisine de celle où une maintenance préventive (voire corrective) a lieu, ou encore sur un autre organe d'un système donnant lieu à une intervention. Exemple : le remplacement et/ ou le meulage de certaines portions de voies. Ce type d'action vise à optimiser les coûts de maintenance, les temps d'intervention et d'utilisation des matériels.

#### 2.4.5 Maintenance imparfaite

Les modèles de maintenance supposent souvent que le composant, après les tâches préventives ou correctives, est "comme neuf"(AGAN *As Good As New*). Mais, dans certains cas, le système n'est pas réellement remis à neuf, après l'entretien préventif.

Il existe une littérature abondante sur les modèles de la maintenance préventive imparfaite [Ben-Daya 2000], [Osaki 2002], [Nakagawa 2005].

Dans notre travail nous modéliserons les effets de la maintenance préventive imparfaite par la réduction de l'âge effectif du composant maintenu, en utilisant un facteur d'ajustement.

La probabilité que le composant soit en état de bon fonctionnement après une

action de maintenance préventive donnée imparfaite est alors :

$$P(X_{k\tau}) = (1 - \alpha)P(X_1) \quad (2.7)$$

où  $\alpha$  est le facteur d'ajustement.

## 2.5 Modélisation de la maintenance

L'utilisation des Modèles Graphiques Probabilistes (MGP) offre un formalisme puissant et intuitif en matière de modélisation, ainsi qu'un ensemble d'outils d'analyses adaptés à l'évaluation des politiques de maintenance. Les travaux de [Corset 2003] et [Celeux 2006] ont abordé la construction pratique d'un MGP dans le contexte d'une étude de sûreté de fonctionnement. Par ailleurs, [Kang 1999] a proposé un MGP statique dédié à la modélisation de la maintenance, en incluant explicitement une couche diagnostic.

Néanmoins, tous ces modèles ne représentent pas une solution générale et complète pour ce problème. En revanche, plus récemment, [Bouillaut 2011] a proposé un environnement générique de modélisation, baptisé "VirMaLab" (Atelier Virtuel de Maintenance).

Comme certaines des approches citées juste avant, cet environnement repose sur une modélisation en MGP Markoviens (MGPM) où le modèle de dégradation du système considéré est codé sous la forme d'une chaîne de Markov ou sous la forme d'un modèle graphique de durée semi-Markovienne. Les couches suivantes de ce modèle décrivent les composantes de diagnostic, de maintenance et d'utilité. Ensuite, l'idée consiste à enrichir ce modèle de dégradation en ajoutant une composante de diagnostic, de maintenance et d'utilité. Il s'agit au final d'un outil d'analyse complet capable de s'adapter aux différentes stratégies de maintenance rencontrées couramment dans l'industrie.

Un des objectifs de notre travail consiste à profiter de cet environnement de modélisation générique et dynamique pour proposer une fonction d'utilité générique pour l'évaluation des politiques de maintenance. Cette fonction doit tenir compte de tout le contexte de la politique de maintenance afin de pouvoir évaluer son utilité.

La partie 2.5.1 est consacrée à la description du modèle VirMaLab. La partie 2.6 aborde le calcul d'utilité permettant l'évaluation d'une politique de maintenance donnée.

### 2.5.1 Description du modèle VirMaLab

Le modèle graphique associé à l'approche VirMaLab repose sur le formalisme MGD introduit dans 1.4.2. Il caractérise les relations de dépendance du vecteur aléatoire  $(\mathbf{X}_t, \mathbf{D}_t, A_t)_{t \geq 1}$  sur deux tranches temporelles comme le montre la figure 2.2. Le modèle VirMaLab est constitué de trois composantes principales dédiées

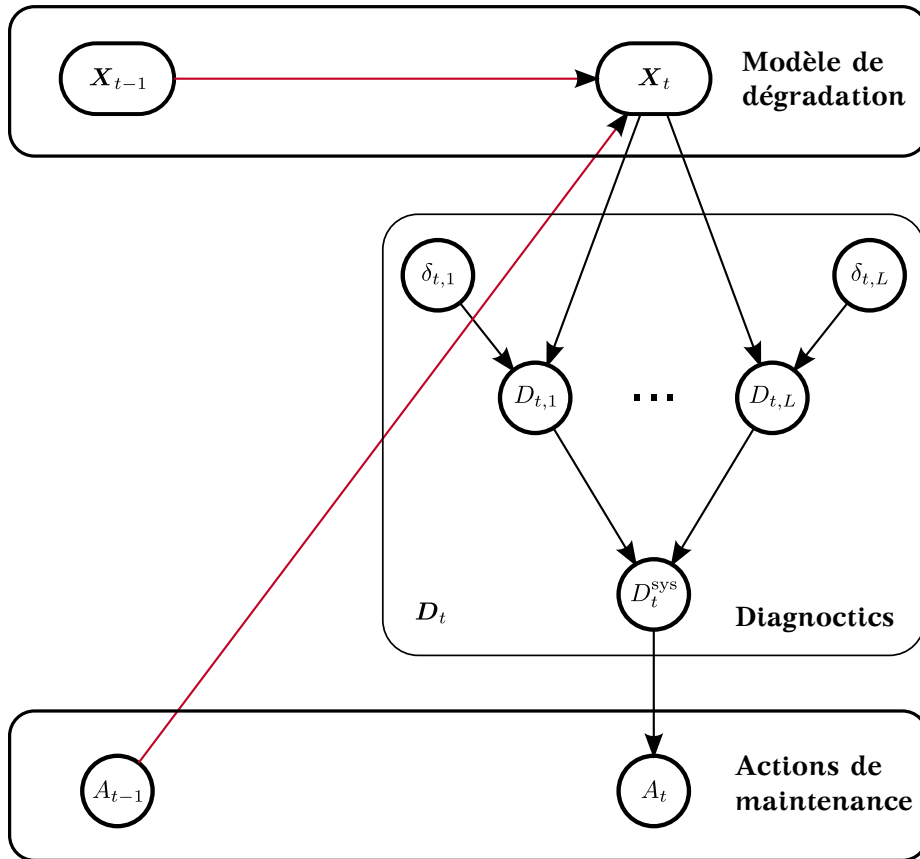


FIGURE 2.2 – Modèle de transition associé au modèle VirMaLab. Pour plus de lisibilité, seul le modèle de transition est représenté sur la figure. Le modèle initial étant identique à celui d'une tranche de temps quelconque, il n'y a pas de confusion possible.

respectivement à l'évolution du système (processus  $(\mathbf{X}_t)_{t \geq 1}$ ), à son diagnostic (processus  $(\mathbf{D}_t)_{t \geq 1}$ ) et à sa maintenance (processus  $(A_t)_{t \geq 1}$ ).

Le vecteur aléatoire  $\mathbf{X}_t$  représente l'état du système à l'instant  $t$ . L'état du système désigne ici toutes les variables permettant d'expliquer son évolution dans le temps. Pour plus de clarté, nous réduisons le vecteur  $\mathbf{X}_t$  à une simple variable



$X_t$ .

Le vecteur aléatoire  $\mathbf{D}_t = ((\delta_{t,\ell}, D_{t,\ell})_{1 \leq \ell \leq L}, D_t^{\text{sys}})$  représente l'ensemble des moyens de diagnostic du système. Le modèle de diagnostic est composé de  $L$  méthodes de diagnostic indépendantes conditionnellement à l'état du système dont les sorties sont les variables aléatoires  $D_{t,1}, \dots, D_{t,L}$ . Chaque variable  $D_{t,\ell}$  représente le résultat du diagnostic de la  $\ell$ -ème méthode à l'instant  $t$ .

Les variables  $\delta_{t,1}, \dots, \delta_{t,L} \in \{0, 1\}$  sont des variables binaires indiquant si chaque méthode de diagnostic est activée à l'instant  $t$ . Pour un système de monitoring en temps réel,  $\delta_{t,j} = 1, \forall t$  et  $i \in \{1, \dots, L\}$  tel que  $\delta_{t,i} = \delta_{T,i}$  pour les auscultations calendaires.

La variable  $D_t^{\text{sys}}$  représente le diagnostic final du système à l'instant  $t$  issu du traitement des résultats des  $L$  méthodes de diagnostic.

La variable  $A_t$  désigne l'action de maintenance sélectionnée à l'instant  $t$ . Dans le modèle VirMaLab considéré dans ces travaux, la décision de maintenance est prise à partir du diagnostic final. Cette action agit ensuite potentiellement sur l'état du système à l'instant suivant. En se référant à la proposition 1.4.1, le vecteur aléatoire associé à l'interface gauche dans le modèle VirMaLab à l'instant  $t$  est  $(X_{t-1}, A_{t-1})$ . Les relations d'indépendances engendrées par la structure graphique du modèle sont les suivantes :

$$X_{t-2}, D_{t-2}, A_{t-2} \perp\!\!\!\perp X_t, D_t, A_t | X_{t-1}, A_{t-1}. \quad (2.8)$$

Ces relations de dépendance conduisent à la définition récursive suivante pour l'interface gauche :

$$P(X_t, A_t) = \begin{cases} P(X_1) \sum_{D_1} P(D_1 | X_1) P(A_1 | D_1) & t = 1 \\ \sum_{\substack{X_{t-1} \\ A_{t-1}}} P(X_{t-1}, A_{t-1}) P(X_t | X_{t-1}, A_{t-1}) \sum_{D_t} P(D_t | X_{t-1}) P(A_t | D_t) & t \geq 2 \end{cases} \quad (2.9)$$

avec pour tout  $t \geq 1$ ,

$$P(D_t) = \prod_{\ell=1}^L [P(\delta_{t,\ell}) P(D_{t,\ell} | X_t, \delta_{t,\ell})] P(D_t^{\text{sys}} | D_{t,1}, \dots, D_{t,L}),$$

et

$$P(A_t | D_t) = P(A_t | D_t^{\text{sys}}).$$

La structure graphique associée à l'approche VirMaLab permet de représenter :

- le processus de dégradation d'un système markovien avec le formalisme MGD;

- un nombre quelconque de méthodes de diagnostic dont l'activation dépend du temps ;
- une politique de traitement des différents résultats de diagnostic ;
- une politique d'action de maintenance.

L'environnement de modélisation VirMaLab possède l'avantage d'être à la fois réaliste et générique. Il s'adapte en effet à de nombreux besoins industriels et couvre un large éventail de problèmes liés à la modélisation de la maintenance.

### 2.5.2 Paramètres du modèle VirMaLab

Les différents paramètres de VirMaLab peuvent être classés en deux catégories : les paramètres descriptifs et les paramètres de maintenance.

La première catégorie correspond aux paramètres caractérisant le fonctionnement du système considéré. Il s'agit donc des éléments associés à l'évolution du système, à la précision des méthodes de diagnostic et aux effets des actions de maintenance. En pratique, ces paramètres se déduisent des Retours d'EXpérience (REX), des avis d'experts, des caractéristiques techniques du matériel utilisé ou encore des référentiels de maintenance. Par conséquent, hormis lors de changements techniques, les paramètres descriptifs sont supposés fixes pour un système donné.

Dans la suite, nous adoptons la notation  $\mathcal{P}^{\text{desc}}$  pour désigner les paramètres descriptifs du modèle VirMaLab.

La seconde catégorie de paramètres décrit la stratégie de maintenance du système. Il s'agit par exemple des choix des instants d'activation des diagnostics  $(\delta_{t,\ell})_{t \geq 1, 1 \leq \ell \leq L}$ , des traitements des diagnostics et de la politique de décisions relativement aux actions de maintenance. Nous désignons par  $\mathcal{P}^{\text{maint}}$  les paramètres de maintenance du modèle VirMaLab. De manière générale, l'objectif d'une telle approche consiste à déterminer  $\mathcal{P}^{\text{maint}}$ , étant donné  $\mathcal{P}^{\text{desc}}$  de façon à optimiser un certain critère. Cependant, dans certaines études, le raisonnement inverse peut être effectué ; à savoir pour une politique de maintenance donnée ; quels sont les paramètres d'utilisation optimales.

Il est alors possible d'utiliser la notation  $\mathcal{P} = (\mathcal{P}^{\text{desc}}, \mathcal{P}^{\text{maint}})$  pour caractériser complètement un modèle VirMaLab, désignant à la fois ses paramètres descriptifs et ses paramètres de maintenance.

## 2.6 Évaluation des politiques de la maintenance

Il s'agit à présent de pouvoir évaluer l'indisponibilité et les coûts de maintenance d'une stratégie donnée. Considérons un modèle VirMaLab, noté  $\mathcal{P} =$

$(\mathcal{P}^{\text{desc}}, \mathcal{P}^{\text{maint}})$ , caractérisé par ses paramètres descriptifs et ses paramètres de maintenance. Nous introduisant donc, en plus des composantes d'état, de diagnostic et de maintenance, des nœuds d'utilité [Howard 1984, Smith 1989].

En pratique, les nœuds d'utilité sont souvent homogènes à un coût ou à un profit. Ils sont munis d'une table indiquant l'utilité conditionnellement aux nœuds parents.

Toutefois, une utilité correspond plus généralement à une grandeur calculable à partir de la loi du vecteur aléatoire considéré et n'a donc pas forcément une connotation économique. Par exemple dans une étude de fiabilité, une utilité peut être associée à un nombre de défauts particuliers ou à un nombre de fausses alarmes pour un appareil de diagnostic. Elle peut aussi quantifier l'intérêt d'une décision d'intervention ou la faisabilité d'une action par rapport à une contrainte donnée.

Supposons qu'à chaque instant  $t$ , le système, les diagnostics et l'action de maintenance engendrent chacun des utilités. Le graphe 2.3 représente le modèle VirMaLab à la tranche  $t$  sur lequel des nœuds d'utilité sont explicitement représentés. Les paragraphes suivants détaillent tout d'abord le calcul des utilités associées aux différentes composantes du modèle à l'instant  $t$ . La construction d'une fonction d'utilité est traitée par la suite.

### 2.6.1 Disponibilité du système

Soient  $(u_{\mathbf{X}_t}^k)_{\{1 \leq k \leq K\}} \in \mathbb{R}$  une suite de potentiels d'utilité associée au vecteur aléatoire  $\mathbf{X}_t$  (à valeur dans  $\mathbb{X}$ ) représentant l'état du système à l'instant  $t$ .

Pour tout  $k = 1, \dots, K$ , la  $k$ -ème utilité associée à  $\mathbf{X}_t$ , notée  $U_{\mathbf{X}_t}^k \in \mathbb{R}$ , est définie par

$$U_{\mathbf{X}_t}^k = \sum_{\mathbf{x} \in \mathbb{X}} u_{\mathbf{X}_t}^k(\mathbf{x}) P(\mathbf{X}_t = \mathbf{x}). \quad (2.10)$$

Le vecteur  $\mathbf{U}_{\mathbf{X}_t} = (U_{\mathbf{X}_t}^1, \dots, U_{\mathbf{X}_t}^K)$  est utilisé afin de désigner les différentes utilités du système à l'instant  $t$ . Lorsque les potentiels d'utilité ne dépendent pas de  $t$ , ces derniers sont dits homogènes, auquel cas l'indice  $t$  est omis et l'écriture  $u_{\mathbf{X}}^1, \dots, u_{\mathbf{X}}^{K_{\mathbf{X}}} \in \mathbb{R}$  est adoptée.

### 2.6.2 Diagnostic

Soit  $u_{\delta_{t,\ell}}^1, \dots, u_{\delta_{t,\ell}}^{K_{\delta_\ell}}$  la suite d'utilités associée à l'activation de la  $\ell$ -ème méthode de diagnostic à l'instant  $t$ . Pour tout  $\ell = 1, \dots, L$ , tout  $k_{\delta_\ell} = 1, \dots, K_{\delta_\ell}$ , la  $k_{\delta_\ell}$ -ème utilité associée à la variable d'activation  $\delta_{t,\ell} \in \{0, 1\}$ , notée  $U_{\delta_{t,\ell}}^{k_{\delta_\ell}}$ , est définie par

$$U_{\delta_{t,\ell}}^{k_{\delta_\ell}} = u_{\delta_{t,\ell}}^{k_{\delta_\ell}} \delta_{t,\ell}. \quad (2.11)$$

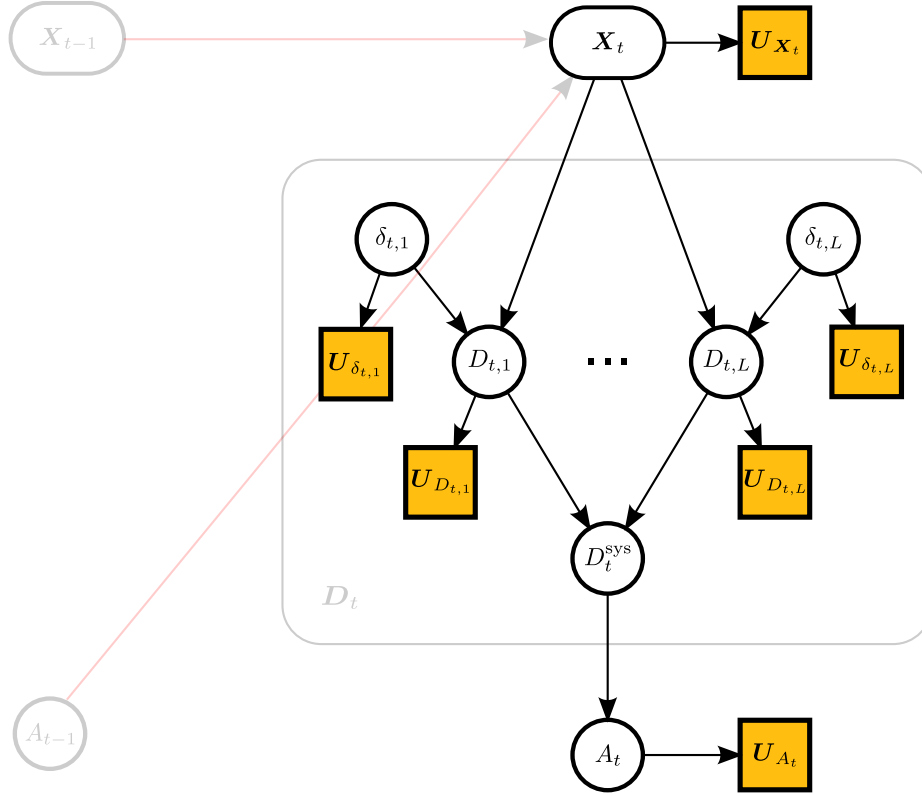


FIGURE 2.3 – Modèle VirMaLab à la tranche  $t$  avec utilités. Les nœuds carrés représentent les utilités des variables parentes associées.

La variable d'activation  $\delta_{t,\ell}$  est une variable binaire qui sert à identifier les méthodes de diagnostic qui doivent s'exécuter à l'instant  $t$ . Il faut noter, qu'un diagnostic peut se produire de manière continue ( $\forall t$ ) ou périodique. L'hypothèse implicite formulée ici consiste donc à fixer une utilité nulle lorsque le diagnostic n'est pas activé et une utilité  $u_{\delta_{t,\ell}}^{k_{\delta_\ell}}$  sinon.

Le vecteur  $\mathbf{U}_{\delta_{t,\ell}} = (U_{\delta_{t,\ell}}^1, \dots, U_{\delta_{t,\ell}}^{K_{\delta_\ell}})$  désigne les utilités liées à l'activation de la  $\ell$ -ème méthode de diagnostic à l'instant  $t$ .

Soit à présent  $u_{D_{t,\ell}}^1, \dots, u_{D_{t,\ell}}^{K_{D_\ell}}$  une suite de potentiels d'utilité associée à la variable aléatoire  $D_{t,\ell}$  représentant le résultat de la  $\ell$ -ème méthode de diagnostic à l'instant  $t$ . Pour tout  $k_{D_\ell} = 1, \dots, K_{D_\ell}$  et tout  $d_\ell \in \mathcal{D}_\ell$ , la  $k_{D_\ell}$ -ème utilité associée à  $D_{t,\ell}$ , notée  $U_{D_{t,\ell}}^{k_{D_\ell}}$ , est définie par

$$U_{D_{t,\ell}}^{k_{D_\ell}} = \sum_{d_\ell \in \mathcal{D}_\ell} u_{D_{t,\ell}}^{k_{D_\ell}}(d_\ell) P(D_{t,\ell} = d_\ell). \quad (2.12)$$

Le vecteur  $\mathbf{U}_{D_{t,\ell}} = (U_{D_{t,\ell}}^1, \dots, U_{D_{t,\ell}}^{K_{D_\ell}})$  désigne les différentes utilités du diagnostic associées à la  $\ell$ -ème méthode à l'instant  $t$ .

De manière synthétique, le vecteur  $\mathbf{U}_{\mathbf{D}_t} = (\mathbf{U}_{\delta_{t,\ell}}, \mathbf{U}_{\mathbf{D}_{t,\ell}})_{1 \leq \ell \leq L}$  contient l'ensemble des utilités liées à la composante diagnostic à l'instant  $t$ .

### 2.6.3 Action de maintenance

Soit  $u_{A_t}^1, \dots, u_{A_t}^{k_A} \in \mathbb{R}^{\mathcal{A}_t}$  une suite de potentiels d'utilité associée à la variable aléatoire  $A_t$  représentant l'action de maintenance à l'instant  $t$ . Pour tout  $a \in \mathcal{A}$ , la  $k_A$ -ème utilité associée à  $A_t$ , notée  $U_{A_t}^{k_A} \in \mathbb{R}$ , est définie par

$$U_{A_t}^{k_A} = \sum_{a \in \mathcal{A}} u_{A_t}^{k_A}(a) P(A_t = d_\ell). \quad (2.13)$$

Le vecteur  $\mathbf{U}_{A_t} = (U_{A_t}^1, \dots, U_{A_t}^{k_A})$  désigne les différentes utilités d'action de maintenance sélectionnée à l'instant  $t$ .

### 2.6.4 Critères d'évaluation d'une politique de maintenance

Il s'agit à présent de proposer un critère permettant d'évaluer, et donc de comparer, chacune des stratégies de maintenance ajustables par les paramètres  $\mathcal{P}^{\text{maint}}$ . La démarche retenue consiste à agréger au sein d'une même fonction les aspects de coûts de maintenance, de diagnostic et de disponibilité. Cela revient alors à combiner les utilités obtenues à chaque instant  $t$  de manière à obtenir un résultat sur un horizon temporel de longueur  $T$ .

La fonction d'utilité définit la manière dont sont combinées les utilités de chaque composante du système.

L'objectif à présent, est de construire une fonction, notée  $\mathcal{U}$ , qui associe à  $\mathcal{P}$  un vecteur  $\mathcal{U}(\mathcal{P}) \in \mathbb{R}$ . Il est important de noter que les calculs d'utilité associés aux équations (2.10), (2.12) et (2.13) nécessitent des calculs d'inférence probabiliste qui dépendent directement des paramètres du modèle  $\mathcal{P}$ . Seules les utilités liées à l'activation des diagnostics s'obtiennent immédiatement à partir des valeurs prises par les variables d'activation  $(\delta_\ell)_{1 \leq \ell \leq L}$ . Autrement dit, l'expression de la fonction d'utilité peut se réécrire en faisant intervenir explicitement les utilités impliquées dans le calcul :

$$\mathcal{U}(\mathcal{P}) = \mathcal{U}((\mathbf{U}_{\mathbf{X}_t}(\mathcal{P}), \mathbf{U}_{\mathbf{D}_t}(\mathcal{P}), \mathbf{U}_{A_t}(\mathcal{P}))_{1 \leq t \leq T}). \quad (2.14)$$

Soit  $\mathcal{S}$  un système multi-composants multi-états comportant  $N$  composants. Nous disposons des distributions de probabilité des nœuds d'états  $P(\mathbf{X}_{i,t})_{1 \leq i \leq N}$ , de diagnostic  $(P(\mathbf{D}_{i,t,\ell}))_{\substack{1 \leq \ell \leq L \\ 1 \leq i \leq N}}$  et de maintenance  $P(A_t)$ , du modèle représentatif du système et représenté dans le figure 2.4.

Pour plus de lisibilité, nous considérons dans la suite du travail qu'il y au

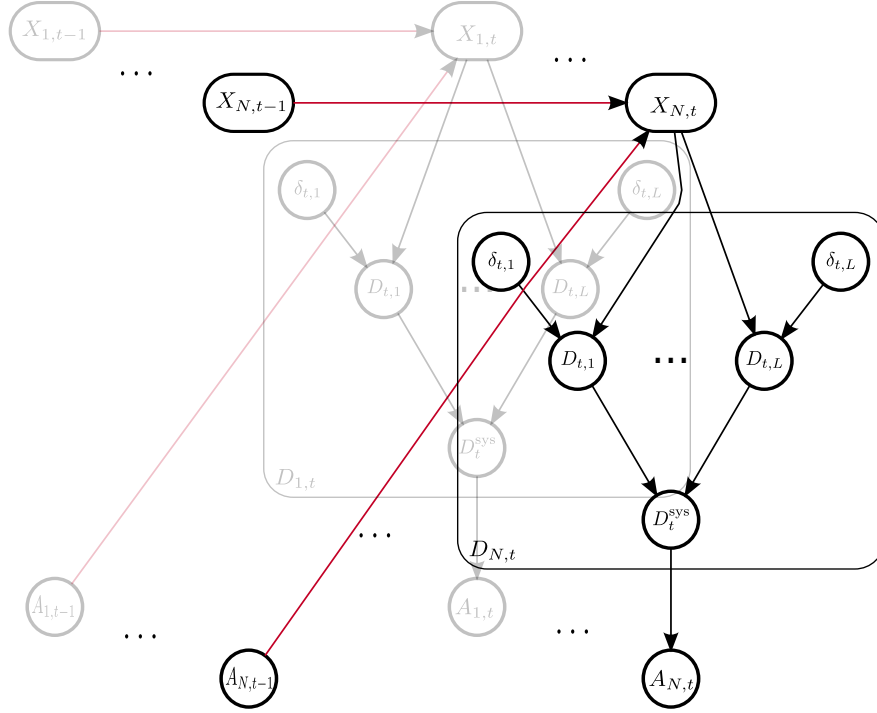


FIGURE 2.4 – Modélisation d'un multi-composants multi-états comportant N composants

plus un seul nœud d'utilité affecté pour un nœud donné, ceci se traduit par  $K = K_{\delta_\ell} = K_{D_\ell} = K_A = 1$ .

#### 2.6.4.1 Mesure de performance au niveau composant

Une bonne planification de la maintenance passe tout d'abord par la prise en compte du coût de l'entretien, ainsi que la disponibilité du composant dans le temps. Comme nous avons l'intention d'examiner l'influence de l'entretien d'un composant sur le fonctionnement entier du système, nous supposons que le composant est inactif au cours de sa période d'entretien et nous introduisons un état supplémentaire des systèmes correspondant à cette phase. Soit, à l'instant  $t$ , l'utilité associée au fonctionnement d'un composant  $i$  :

$$u_t^i = U_{X_{i,t}} + \sum_{\ell=1}^L (U_{\delta_{i,\ell,t}} + U_{D_{i,t,\ell}}) \quad (2.15)$$

Comme  $S$  est un système à composants multi-états nous pouvons alors récrire le potentiel d'utilité  $u_{X_t}$  sous la forme d'un vecteur,  $u_{X_t} = [u_{X_t}^1 \dots u_{X_t}^E]$ .  $|E|$  correspond au nombre d'états que peut prendre un composant. D'où cette écriture de

$U_{X_{i,t}}$  :

$$U_{X_{i,t}} = \sum_{e \in E} u_{X_t}^e P(X_t = e) \quad (2.16)$$

Il est à rappeler qu'un état  $e$  peut représenter plusieurs types d'états et de situations que nous définissons selon les besoins. Il peut correspondre par exemple à :

- état de bon fonctionnement
- état dégradé
- état d'arrêt pour réparation
- état de panne totale

Selon les exigences de sûreté, de productivité ou économiques, les potentiels d'utilités doivent être calibrés adéquatement pour donner une estimation véritable de des coûts engendrés et par conséquent répondre au mieux au besoin de l'optimisation de la maintenance.

#### 2.6.4.2 Mesure de la performance d'une stratégie de maintenance au niveau système

Dans cette partie, une estimation globale de la politique de maintenance appliquée au système  $\mathcal{S}$  (multi-composants, multi-états) est donnée [Ayadi 2011]. La fonction d'utilité globale à un instant  $t$  est définie dans (2.17) par une agrégation d'utilités.

$$u_t = \sum_{i=1}^N u_t^i + \sum_{s \in \mathbb{X}^{\mathcal{S}}} u_{X_t^{\mathcal{S}}} P(X_t^{\mathcal{S}} = s) + U_{A_t} \quad (2.17)$$

où  $X_t^{\mathcal{S}}$  et  $\mathbb{X}^{\mathcal{S}}$  désignent respectivement l'état global à l'instant  $t$  et l'ensemble des états du système  $\mathcal{S}$ . L'utilité est communément appelée "Fitness".  $(u_{X_t^{\mathcal{S}}})_{s \in \mathbb{X}^{\mathcal{S}}}$  correspond au potentiel d'utilités associés à l'état du système. L'utilité moyenne par unité de temps sera alors :

$$u = \sum_{t=1}^T \frac{u_t}{T} \quad (2.18)$$

D'après les équations (2.17) et (2.15), nous distinguons principalement à chaque fois deux facteurs : un facteur traduisant l'indisponibilité du système, un autre le diagnostic et un dernier concernant la maintenance. Nous reformulons, en conséquence, ci-dessous l'équation (2.17) en mettant en intégrant les facteurs cités.

$$u_t = \sum_{i=1}^N U_{X_{i,t}} + \sum_{s \in \mathbb{X}^{\mathcal{S}}} u_{X_t^{\mathcal{S}}} P(X_t^{\mathcal{S}} = s) + \sum_{i=1}^N \sum_{\ell=1}^L (U_{\delta_{i,\ell,t}} + U_{D_{i,t,\ell}}) + U_{A_t} \quad (2.19)$$

Cette réécriture est importante dans la mesure où elle dégage clairement les principaux critères à optimiser dans une politique de maintenance.

Dans ce qui suit, nous proposons un exemple simple permettant d'illustrer les notions précédemment introduites.

### Exemple 2.6.1

*Le système considéré est une machine de production composée de deux composants, nommés B et C, montés en série.*

**État des composants :** Chaque composant possède cinq états de fonctionnement : neuf, dég1, dég2, panne et Rép. Les états de dégradation sont ordonnés par gravité croissante ce qui fait de dég2 un état de dégradation critique. Rép représente l'état en réparation. Notons également que la machine de production dans son ensemble, qu'on note  $S$ , peut avoir trois états : OK, panne et Rép.

**État du système :**  $S$  est en panne si B et C sont en état dég2 ou si un des deux est en état panne. Il suffit qu'un des composants soit dans un état de réparation pour que  $S$  soit en état de réparation.

**Diagnostic :** Nous supposons que le système possède deux méthodes de diagnostic : une méthode 1, représentée par la variable  $D_{t,1}$  et une méthode 2, représentée par la variable  $D_{t,2}$ . Ces deux méthodes sont subordonnées respectivement par deux variables d'activations  $\delta_{t,1}$  et  $\delta_{t,2}$ .

**Maintenance :** La stratégie de maintenance appliquée à ce système est une maintenance de type mixte. En fait,  $D_{t,1}$  est activée à chaque pas de temps pour surveiller l'évolution des états des composants et déclencher une action de maintenance si le système est en panne.  $D_{t,2}$  est activé périodiquement.

La variable de maintenance possède quatre états : ne rien faire "rien", réparer B "RépB" (respectivement réparer C "RépC") ou réparer B et C "RépBC".

L'objectif à présent, est d'évaluer la stratégie de maintenance du système en un instant  $t$  donné. Pour plus de simplicité, dans cet exemple la notion d'utilité est homogène à une grandeur économique dont l'unité n'est pas précisée. Une utilité négative correspond à un coût, tandis qu'une utilité positive correspond à un gain.

Dans un premier temps, nous supposons que la qualité de la production dépend de l'état dans lequel se trouve chacun des composants. Plus précisément, des bénéfices sont réalisés tant que ces derniers sont neufs ou peu dégradés. En revanche, dès lors que l'état dég2 est atteint, des pertes sont occasionnées.

Numériquement, le potentiel d'utilité de B (respectivement de C) relativement à son état est noté  $u_{x_B}$  (respectivement  $u_{x_C}$ ). Les tableaux 2.1 présentent les états, les potentiels d'utilités et le résultat du calcul de l'utilité correspondant aux composants B et C. Le tableau 2.2 récapitule l'état du système  $S$ , les potentiels d'utilités (coûts) associés à ses états et le coût résultant de l'indisponibilité. En examinant



B				C			
	$P(X_{B,t})$	$u_{X_B}$	$U_{X_{B,t}}$		$P(X_{C,t})$	$u_{C_B}$	$U_{X_{C,t}}$
neuf	0.63	0	-0.085	neuf	0	0	0
dég1	0.25	0		dég1	0	0	
dég2	0.07	-0.5		dég2	0	-0.5	
panne	0.05	-1		panne	0	-1	
Rép	0	0		Rép	1	0	

TABLE 2.1 – Tableau récapitulatif du calcul de l'utilité associée aux composants B et C

$\mathcal{S}$			
	$P(X_t^{\mathcal{S}})$	$u_{X_t^{\mathcal{S}}}$	$U_{X_t^{\mathcal{S}}}$
Ok	0	+1	0
panne	0	-1	
Rép	1	0	

TABLE 2.2 – Calcul du coût de l'indisponibilité du système

ces tableaux, il est clair que la production à cet instant  $t$  est en arrêt pour cause de maintenance.

Le tableau 2.3 résume les actions de maintenance et leurs probabilités d'existence

Maintenance de $\mathcal{S}$			
	$P(A_t)$	$u_{A_t}$	$U_{A_t}$
Rien	0	0	-4
RépB	0	-4	
RépC	1	-4	
RépBC	0	-7	

TABLE 2.3 – Tableau récapitulatif des actions de maintenance, leurs potentiel d'utilité et le coût résultant de la maintenance

à l'instant actuel  $t$ , les potentiels d'utilités relatifs aux coûts des actions de maintenance et enfin le résultat du calcul du coût actuel de la maintenance. Étant donnée

Utilité du diagnostic pour B			
	Probabilité d'existence	Potentiel d'utilité	$U_{D_{B,t,1}} = U_{\delta_{B,t,1}}, U_{D_{B,t,1}}$
$\delta_{B,t,1}$	1	$10^{-6}$	$10^{-6}$
$D_{B,t,1}$	1	$10^{-5}$	$10^{-5}$

TABLE 2.4 – Récapitulatif du coût du diagnostic pour le composant B

que  $\forall t$ ,  $D_{t,1}$  est activée, on a alors  $\delta_{B,t,1} = \delta_{C,t,1} = 1$ . Nous supposons que la deuxième méthode de diagnostic  $D_{t,2}$ , qui est périodique, n'est pas activée ni pour B ni pour C à l'instant  $t$ . Le calcul de l'utilité du diagnostic se résume alors au calcul du coût du diagnostic et de l'activation de  $D_{t,1}$ , appliquée au deux composants.

Le tableau 2.4 résume le calcul de l'utilité du diagnostic pour le composant B. Le même calcul est valable pour le composant C.

En se basant sur les calculs numérique ci-dessus présentés, nous concluons que l'utilité globale de la politique de maintenance appliquée à la machine de production à l'instant  $t$  est égale à :

$$U_{X_{B,t}} + U_{X_{C,t}} + U_{D_{B,t,1}} + U_{D_{C,t,1}} + U_{X_t^S} + U_{A_t} = -4.085022 \simeq -4.09 \quad (2.20)$$

L'estimation de l'utilité à un instant donné  $t$  nous renseigne sur l'ordre de grandeur de l'utilité mais ne nous informe pas sur l'impact de la politique de maintenance sur le système au long terme. C'est pourquoi, il est indispensable d'évaluer l'utilité globale moyenne sur une période de temps  $T$ . Il s'agit d'abord de fixer un horizon temporel de fonctionnement du système  $T$  avant renouvellement et ensuite calculer l'utilité à chaque instant  $t \in [0, T]$  et faire la moyenne. L'utilité globale moyenne est donc fonction des paramètres de maintenance et du temps de l'étude. Ceci laisse entrevoir un réglage important de variables afin d'établir le meilleur plan de maintenance promettant une utilité globale moyenne optimale.

## 2.7 Conclusion

Les développements présentés dans ce chapitre ont abouti à la formalisation d'une fonction d'utilité pour la quantification des stratégies de maintenance complexes appliquées à des systèmes multi-composants.

Les travaux réalisés dans le cadre de cette thèse avaient, tout d'abord, pour objectif la définition d'une approche de modélisation de la dégradation applicable à un large panel de systèmes puis l'élaboration d'une fonction d'utilité qui évalue fidèlement les stratégies de maintenance. La fonction d'utilité proposée

prend en compte les interactions entre les différents paramètres de maintenance, de diagnostic, de l'état du système ainsi que de l'environnement de l'étude. Par conséquent, afin de trouver la meilleur utilité globale qui soit, d'une politique de maintenance préventive appliquée à un système donné fonctionnant dans un contexte particulier, il faudrait réussir à choisir les paramètres de maintenance et de diagnostic adéquats, ce qui n'est pas toujours facile. Cette difficulté révèle la nécessité de disposer de méthodes d'optimisation afin de déterminer les paramètres optimaux d'une stratégie de maintenance.



# Les métaheuristiques : état de l'art

---

## Sommaire

<b>3.1 Introduction</b>	<b>63</b>
<b>3.2 Les méthodes de résolution</b>	<b>64</b>
<b>3.3 L'optimisation multiobjectif</b>	<b>70</b>
<b>3.4 Transformation du POM en un problème mono-objectif</b>	<b>74</b>
<b>3.5 Approches non Pareto</b>	<b>82</b>
<b>3.6 Approches Pareto</b>	<b>83</b>
<b>3.7 Conclusion</b>	<b>86</b>

---

## 3.1 Introduction

Souvent, la résolution des problèmes réels passe par l'optimisation à la fois d'un ou plusieurs critères (objectifs) afin de trouver une solution satisfaisante.

Toutefois, trouver cette solution satisfaisante, qui réconcilie plusieurs critères n'est pas sans difficultés. En effet, les critères à optimiser sont souvent antagonistes : l'amélioration d'un objectif provoque la détérioration d'un autre objectif.

Par conséquent, le résultat final de l'optimisation n'est plus donné par une solution unique mais plutôt par un ensemble de solutions de compromis entre les différents critères à optimiser.

Par ailleurs, ce chapitre, présente essentiellement l'état de l'art de l'optimisation évolutionnaire multi-objectif. Dans un premier temps, nous passons rapidement en revue les approches générales de résolution de l'optimisation. Par la suite, nous définissons le problème d'optimisation multiobjectif, son vocabulaire et les notions relatives. Nous finissons ce chapitre par la description de différents types de métaheuristiques.

## 3.2 Les méthodes de résolution

Les méthodes de l'optimisation de la maintenance sont très variées. Le choix d'une méthode de résolution par rapport à une autre dépend essentiellement de la nature du problème à résoudre, sa complexité et les données disponibles.

Nombreux sont les études menées sur les approches d'optimisation en général [Allaire 2005], [Corriou 2010] et sur les méthodes d'optimisation de la maintenance et de la fiabilité en particulier [Dekker 1996], [Campbell 2001], [Pham 2006], [Nakagawa 2010], [Kolowrocki 2011].

De manière générale, il existe deux classes de méthodes de résolution :

- Les méthodes exactes : garantissent l'intégralité de la résolution avec un temps de calcul polynomial, souvent inacceptable si la taille du problème en question dépasse une certaine limite de grandeur. En théorie, ces méthodes aboutissent à des solutions optimales du problème.
- Les méthodes approchées (heuristiques/métaheuristiques) : donnent des solutions proches de l'optimal en un temps, souvent, plus raisonnable que celui des méthodes exactes. Cependant, la qualité de ces solutions n'est pas garantie par la théorie.

### 3.2.1 Méthodes exactes

Dans les méthodes exactes, le temps de calcul est étroitement lié à la taille du problème. Ceci, rend, inapproprié, leur utilisation dans la résolution des problèmes de grande taille. Les méthodes exactes sont alors essentiellement utilisées pour résoudre des problèmes de petites taille. Elles sont basées soit sur une résolution algorithmique ou analytique, soit sur une énumération exhaustive de toutes les solutions possibles. Elles s'appliquent donc aux problèmes qui peuvent être résolus de façon optimale et rapide.

#### 3.2.1.1 Les méthodes analytiques

Les méthodes analytiques (numériques) sont basées sur l'existence d'équations ou de systèmes d'équations, dérivables, linéaires ou non linéaires. L'idée est de rechercher l'extremum en déterminant les points de pente nulle dans toutes les directions.

Parmi les méthodes analytiques les plus connues, on trouve les méthodes du Gradient [Brown 1994]. Le principe de ces méthodes consiste à créer une suite décroissante qui convergera vers le minimum recherché de la fonction objectif.

Ces méthodes peuvent rencontrer un certain nombre de problèmes comme l'existence des dérivées et la convergence lorsque l'extremum de la fonction se trouve

au fond d'une vallée étroite ou dans le cas de fonctions non lisses. Ces méthodes s'appliquent localement, les extremums qu'elles atteignent sont optimaux au voisinage du point de départ

### 3.2.1.2 Les méthodes algorithmiques

#### La programmation linéaire

La programmation linéaire [Guéret 2000] est connue comme l'une des techniques classiques en recherche opérationnelle. Elle est utilisée que lorsque la fonction objectif et les contraintes sont des fonctions linéaires de variables. Le principe de résolution d'un problème d'optimisation linéaire est de déterminer l'optimum d'une fonction objectif  $F$  tout en respectant un certain nombre de contraintes. Soit  $m$  le nombre de contrainte d'égalité et d'inégalité et  $n$  le nombre de contraintes simple de positivité. La formulation général d'un problème de programmation linéaire est la suivante :

$$\text{Max ou Min } F = \sum_{j=1}^n c_j x_j \quad (3.1)$$

$$\sum_{j=1}^n a_{ij} x_j \leq \text{ou} \geq b_i, \forall i = 1 \dots m \quad (3.2)$$

$$x_j \geq 0, \forall j = 1 \dots n \quad (3.3)$$

où,  $c_j$  et  $a_{ij}$  sont les coefficient d'utilité (coût) de la variable  $X_j$ , respectivement, dans la fonction objectif et dans la contrainte  $i$ .

Parmi les algorithmes qui permettent de résoudre ce type de problèmes nous citons l'algorithme du Simplex [Dantzig 1987]. Cet algorithme a été adapté sous une forme matricielle pour faciliter sa programmation informatique. Cette déclinaison de l'algorithme décrit de manière algébrique, le déroulement des opérations soumis à la résolution des systèmes d'équations linéaires.

De nombreuses recherches ont été menées et sont encore menées dans l'application de la programmation linéaire à des cas d'optimisation de la maintenance. On peut citer par exemple l'ouvrage de [Kolowrocki 2011] qui présente des exemples de traitement de problèmes d'optimisation en fiabilité, disponibilité et sûreté de fonctionnement des systèmes techniques complexes.

#### La théorie des Graphes et méthodes arborescentes

La Théorie des Graphes [Sakarovitch 1984], [Lacomme 2003] comme les méthodes arborescentes traitent les problèmes complexes à partir d'une exploration intelligente de l'espace de recherche. Ces méthodes permettent la construction d'une arborescence : les sommets représentent les sous-problèmes et les arcs issus d'un même sommet représentent, chacun, une décomposition possible du

problème, situé au sommet de l'arbre, en sous-problèmes de taille réduite. Cette arborescence est explorée de manière à exclure les branches sans solutions réalisables ou ceux n'amenant pas à des solutions meilleures que la solution courante.

Ce type de méthodes s'adaptent très bien à des problèmes de taille moyenne. En revanche, pour les problèmes de grande taille, la durée d'exécution peut vite devenir excessive.

### **La programmation dynamique**

La programmation dynamique est une méthodologie de construction d'algorithmes, à vocation générale, pour la résolution de certains problèmes d'optimisation. Cependant la taille des problèmes qu'elle permet de traiter peut être limitée. Elle est surtout utilisée en optimisation combinatoire .

La programmation dynamique repose sur le principe d'optimalité de Bellman [Bellman 1971] qui stipule que “ toute politique optimale est composée de sous-politiques optimales”. Ce principe permet une résolution du problème d'optimisation par étapes (sous problèmes) : Chaque étape correspond à la résolution de façon optimale d'un sous-problème, tout en tenant compte des informations obtenues aux étapes précédentes. Les sous-problèmes peuvent être en interaction, ce qui signifie qu'un sous-problèmes peut être utilisé dans la solution de deux sous-problèmes différents. La résolution des sous-problèmes se fait de manière ascendante : on commence par calculer les solutions des plus petits sous-problèmes pour ensuite déduire les solutions des sous-problèmes de plus en plus grands. Cette démarche, demande surtout une formulation par récurrence de la fonction objectif. Un exemple de ce type de problème est le plus court chemin entre deux sommets d'un graphe.

L'inconvénient de telles méthodes est que leur application aux problèmes réels mène souvent à une simplification forte du problème. Par conséquent, le résultat final peut être loin de l'optimum réel, bien qu'il soit exact. Pour l'optimisation des stratégies de maintenance, dont la complexité peut être très grande en terme d'espace des solutions possibles, et qui de plus est soumis à des perturbations aléatoires, l'utilisation de ces différentes méthodes ne semble pas être la plus appropriée.

### **3.2.2 Méthodes approchées**

Souvent, dans les problèmes de complexité exponentielle, il est impossible (en temps de calcul) de réaliser une exploration totale ou même partielle de l'arborescence des solutions. Les méthodes approchées (appelées aussi heuristiques /métaheuristiques) peuvent pallier ce problème en cherchant à déterminer la bonne solution en termes de qualité et de résultats avec un temps de calcul raisonnable



et en exploitant au mieux la structure du problème [Lopez 2001].

Par ailleurs, ces méthodes présentent également d'autres avantages bien qu'il ne garantissent pas l'optimalité de la solution.

- L'adaptation : les méthodes heuristiques sont applicables à n'importe quelle classe de problèmes, contrairement aux méthodes exactes qui ne peuvent pas opérer dans certaines applications pratiques ( i.e nombre de variables et de contraintes important, divergence des objectifs à optimiser, imprécision des des données).
- La simplicité : les méthodes heuristiques sont généralement faciles à appliquer car elles se basent sur des principes simples et ne nécessitent pas souvent une modélisation mathématique du problème.
- La flexibilité : les méthodes heuristiques peuvent facilement se conjuguer avec d'autres méthodes, formant ainsi une méthode hybride. Cette flexibilité permet souvent d'améliorer significativement la qualité des solutions.

Vue les avantages que présentent ces méthodes, ces dernières s'avèrent être bien adaptées à l'optimisation des stratégies de maintenance. Ce type de méthodes sera utilisé dans la suite de ce travail.

Les méthodes heuristiques réunissent principalement les méthodes constructives, les approches de recherche par voisinage et les méthodes aveugles ou méta-heuristiques. Les méthodes constructives fonctionnent de façon itérative. Elles démarrent à partir d'une solution vide et puis au fil des itérations génèrent de nouvelles solutions en les complétant avec des nouvelles composantes de manière incrémentale. En revanche, les méthodes par voisinage partent d'une ou plusieurs solutions qu'elles tentent à améliorer en cherchant de nouvelles solutions dans un voisinage défini.

Soit  $\mathcal{X}$  l'ensemble des solutions admissibles du problème traité (nous supposons que cet ensemble est fini).  $F$  est définie comme la fonction d'évaluation des solutions admissibles. La résolution optimale d'un problème d'optimisation combinatoire (minimisation) en respectant les contraintes qui lui sont associées consiste à chercher une solution  $X^* \in \mathcal{X}$  tel que :

$$F(X^*) = \min_{X \in \mathcal{X}} F(X) \quad (3.4)$$

### 3.2.2.1 Les méthodes constructives

L'approche constructive construit de façon incrémentale une solution  $X = (x_1, x_2, \dots, x_n)$ , variable après l'autre, en s'interdisant de modifier les variables déjà affectées.

L'objectif est donc de diminuer progressivement la taille de l'espace de recherche

en fixant à chaque étape la valeur d'une variable du problème. Ceci revient à trouver un sous ensemble  $\mathcal{X}^k \subseteq \mathcal{X}$  de taille toujours plus petite.

Cette approche conduit à une ou plusieurs solutions optimales lorsque chaque sous ensemble  $\mathcal{X}^k$  fournie au moins une solution optimale  $s^* \in \mathcal{X}$ . Elle est généralement difficile à réaliser sauf si la construction de la solution est statique et ne remet pas en question les choix antérieurs.

Les méthodes constructives sont rapides et simples mais fournissent des solutions de qualité médiocre car elle ne prennent pas en compte le contexte globale du problème lors de la recherche des solutions. En effet, à chaque étape, la solution courante est complétée de façon optimale sans considération des conséquences obtenues au niveau de la solution finale.

La plupart des méthodes constructives sont de type glouton : la solution initial, vide, est complétée, étape par étape, avec des composants de solutions jusqu'à l'obtention d'une solution complète. La manière dont la solution est construite peut être aléatoire ou heuristique.

Une méthode heuristique produit généralement de meilleurs résultats car à chaque étape, la composante à ajouter est choisie sur la base d'un critère à optimiser (coût à minimiser, respectivement un gain à maximiser). Ainsi, La performance des algorithmes gloutons dépend étroitement de la pertinence de l'heuristique utilisée et de leur capacité à exploiter les connaissances du problème.

Le principal inconvénient des méthodes gloutons est qu'elles sont considérées comme étant «à courte vue» et «non récupérable». En revanche, elles sont idéales pour les problèmes à «sous-structure optimale», et sont les mieux adaptés pour des problèmes simples. Il est important, cependant, de noter que l'algorithme glouton peut être utilisé comme un algorithme de sélection ou un algorithme *branch and bounds*.

### 3.2.2.2 Les méthodes de recherche par un voisinage

Les algorithmes de recherche par voisinage explorent l'espace de solutions, de manière itérative, en partant d'une solution initiale acceptable, choisie arbitrairement ou avec une heuristique.

Le voisinage d'une solution  $X$  est l'ensemble des solutions qui peuvent être érigées à partir de  $X$  en un seul "mouvement" : par exemple, le changement de valeur d'une variable de  $X$ .

A chaque itération, le principe de ces méthodes consiste à chercher dans le voisinage une solution qui améliore la qualité de la solution courante. L'algorithme

s'arrête lorsque la condition d'arrêt est vérifiée. Les conditions d'arrêt peuvent être un nombre d'itérations maximal, un seuil d'évaluation de la qualité de la solution courante ou bien la non amélioration de la solution courante.

Parmi les méthodes de recherche par voisinage, on peut citer l'algorithme du Grimpeur "Hill Climbing" (ou la descente stochastique). Cet algorithme se termine avec un optimum local. L'optimum local est la meilleure solution parmi les solutions du voisinage, contrairement à l'optimum global qui est la meilleure solution parmi toutes les solutions possibles. Plusieurs améliorations ont été proposées, dans la littérature, pour essayer d'échapper aux optimums locaux.

Dans les sections suivantes, quelques unes de ces améliorations comme la recherche tabou et le recuit simulé seront présentées. Pour faire une bonne recherche, ces approches heuristiques doivent établir un équilibre entre l'exploration de l'espace de recherche (zones de l'espace de solutions non encore explorées), l'intensification, visant à exploiter l'information déjà récoltée et l'apprentissage, basé sur la mémoire et qui permet de ne tenir compte que des zones où l'optimum global peut de se trouver.

### 3.2.2.3 Les méthodes aveugles ou métaheuristiques

Les métaheuristiques sont des méthodes stochastiques qui opèrent de manière itérative. Un même schéma d'opérations est reproduit un certain nombre de fois au cours de l'optimisation, sans faire appel au calcul du gradient de la fonction objectif.

Ces méthodes sont réputées pour leur efficacité pour éviter les optimums locaux. En effet, d'une part, elles sont souvent conduites par un ensemble de points, appelé population et d'autre part, elles peuvent accepter des dégradations de la solution optimale courante au cours des itérations. Elle sont capable d'optimiser un problème à partir d'un nombre minimal d'informations mais sans aucune garantie quant à l'optimalité de la meilleure solution trouvée.

Les métaheuristiques sont surtout connues pour résoudre les problèmes à variables discrètes mais font de plus en plus l'objet d'adaptations aux problèmes à variables continues.

Pour l'optimisation d'une politique de maintenance des systèmes industriels, dont la complexité peut être très grande en terme de solutions possibles, et qui de plus est soumis à des perturbations aléatoires, l'utilisation de ces différentes méthodes semble être la plus appropriée.

### 3.3 L'optimisation multiobjectif

La naissance de l'optimisation multiobjectif remonte à un ouvrage de W. Pareto en 1906 sur l'économie politique. Son application aux problèmes d'ingénierie a été au début longtemps limitée à cause de son aspect mathématique important. Cependant avec l'apparition des métaheuristiques, l'optimisation multiobjectif a pris un abord plus facile.

#### 3.3.1 Formulation générale d'un problème d'optimisation multiobjectif

La plupart des problèmes réels requièrent l'optimisation simultanée de plusieurs objectifs. Contrairement à l'optimisation mono-critère, la solution d'un Problème d'Optimisation MultiObjectif (POM) est rarement unique. Elle est constituée d'un assortiment de solutions, représentant l'ensemble des meilleurs compromis entre les objectifs.

Si l'on considère un problème qui prend en compte  $k$  fonctions objectifs,  $m$  contraintes d'inégalité et  $p$  contraintes d'égalité, le problème peut être formalisé comme suit :

$$\left\{ \begin{array}{l} \text{Minimiser} \\ Y = f(X) = (f_1(X), f_2(X), \dots, f_k(X)) \quad k \geq 2 \\ \text{s.c} \\ g(X) = g_i(X) \leq 0 \quad i = 1, 2, \dots, m \\ h(X) = h_j(X) = 0 \quad j = 1, 2, \dots, p \end{array} \right.$$

tel que  $X = (x_1, x_2, \dots, x_n) \in \mathcal{X}^n$  est le vecteur représentant les variables de décisions et  $Y = (y_1, y_2, \dots, y_k) \in \mathcal{Y}^k$  est le vecteur des objectifs.  $\mathcal{X}^n$  et  $\mathcal{Y}^k$  représentent respectivement l'espace de décision de dimension  $n$  (appelé aussi espace de solutions) et l'espace d'objectifs de dimension  $k$  (voir figure 3.1).

Les contraintes  $g(X) \leq 0$  et  $h(X) = 0$  déterminent le domaine  $\mathcal{X}_r^n$  des solutions réalisables dans l'espace décisionnel tel que :

$$\mathcal{X}_r = \{X \in \mathcal{X}^n | g(X) \leq 0, h(X) = 0\} \quad (3.5)$$

A chaque solution  $X_r \in \mathcal{X}_r^n$  est associé un vecteur objectif  $Y_r \in \mathcal{Y}^k$  sur la base d'un vecteur de fonctions  $f : \mathcal{X}_r^n \rightarrow \mathcal{Y}^k$ , où  $\mathcal{Y}^k = f(\mathcal{X}_r^n)$  représente l'ensemble des points réalisables dans l'espace d'objectifs.

Le but de l'optimisation multiobjectif est donc de minimiser un ensemble de fonctions objectif sans trop dégrader les valeurs des optimums obtenus, par une

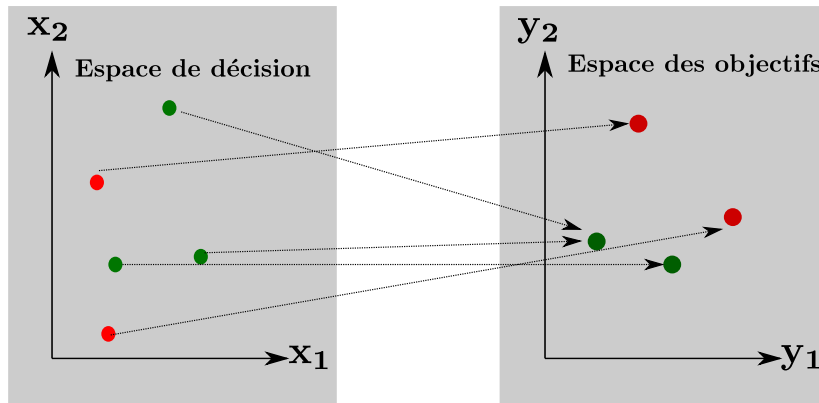


FIGURE 3.1 – Espace de décision et espace d'objectif d'un problème d'optimisation multiobjectif. Exemple d'un problème d'optimisation biobjectif (deux variables de décisions et deux fonctions objectifs)

optimisation mono-objectif effectuée objectif par objectif.

Sauf mention contraire explicite et sans munir à la généralité des approches, toutes les énoncés et définitions seront donnés dans le cadre d'un problème de minimisation.

### 3.3.2 Multiplicité des solutions

La résolution d'un POM ne donne pas une solution unique mais plusieurs solutions possibles. Ceci est dû au fait que, souvent, les POMs, présentent des objectifs antagonistes : l'amélioration d'un objectif entraîne la dégradation de l'autre.

Les solutions résultantes d'un POM sont alors des solutions de compromis.

### 3.3.3 Dominance et front de Pareto

Comme la solution optimale est une multitude de points de  $\mathcal{R}^n$ , il est primordial pour identifier ces meilleurs compromis, de définir une relation d'ordre entre ces éléments.

Dans le cas des POMs, cette relation est appelée relation de dominance et de son existence permet de limiter l'ensemble des solutions de compromis appartenant à l'espace de décision  $\mathcal{X}^n$ .

Il existe différents types de relations de dominance [Collette 2004], qui permettent d'avoir suffisamment de degrés de liberté dans le choix de la relation qui reproduit au mieux le comportement d'un décideur. Mais la plus célèbre et la plus utilisée

est la dominance au sens de Pareto. Elle est définie par :

**Définition 3.3.1 (La relation de dominance au sens de Pareto)**

Le vecteur  $X$  domine le vecteur  $X'$  si  $X$  est au moins aussi bon que  $X'$  dans tous les objectifs, et,  $X$  est strictement meilleur que  $X'$  dans au moins un objectif. Ce qui peut également s'écrire :  $X \in \mathcal{X}^n$  domine  $X' \in \mathcal{X}^n$ , noté  $X \succ X'$  ssi

$$\forall i \in \{1, 2, \dots, k\} f_i(X) \leq f_i(X') \text{ et } \exists j \in \{1, 2, \dots, k\} \text{ tel que } f_j(X) < f_j(X')$$

La figure suivante illustre des solutions dominées et non dominées dans un POM de minimisation.

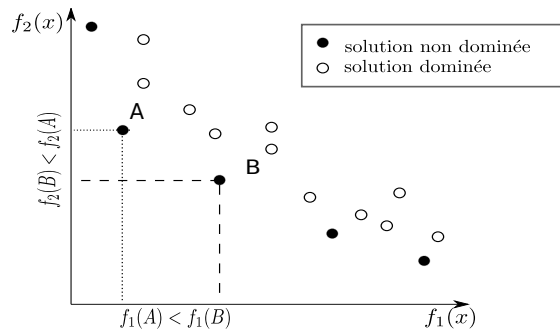


FIGURE 3.2 – Exemple de dominance dans un problème nécessitant la minimisation de deux objectifs ( $f_1$  et  $f_2$ ). Les points A et B sont non dominés.

Afin de définir formellement la notion de dominance au sens de Pareto, les relations  $=$  et  $\leq$ ,  $\geq$  usuelles sont étendues aux vecteurs. Ces relations sont définies pour un problème de minimisation. D'autres définitions peuvent également être introduites :

**Définition 3.3.2 (Dominance faible)**

$X \in \mathcal{X}^n$  est dit faiblement non dominé s'il  $\nexists X' \in \mathcal{X}^n$  tel que

$$f_i(X') < f_i(X) \quad \forall i = \{1, 2, \dots, k\}$$

**Définition 3.3.3 (Dominance forte)**

$X \in \mathcal{X}^n$  domine fortement (strictement)  $X' \in \mathcal{X}^n$  ssi

$$\forall i \in \{1, 2, \dots, k\} f_i(X) < f_i(X')$$

**Définition 3.3.4 (Solution pareto optimale)**

Une solution  $X \in \mathcal{X}^n$  est Pareto optimale ssi  $\forall X' \in \mathcal{X}^n, X' \not\succ X$

**Définition 3.3.5 (Ensemble Pareto optimal, voir figure 3.3))**

Les solutions Pareto optimales forment l'ensemble Pareto optimal, noté  $\mathcal{X}_p^n$ .

$$\mathcal{X}_p^n = \{X \in \mathcal{X}^n \mid \nexists X' \in \mathcal{X}^n, X' \succ X\}$$

**Définition 3.3.6 (Front de Pareto, voir figure 3.3)**

L'image de l'ensemble Pareto optimal dans l'espace des objectifs forme ce qu'on appelle le front de Pareto, noté  $\mathcal{Y}_p^k$ .  $\mathcal{Y}_p^k = \{f(X) \mid X \in \mathcal{X}_p^n\}$

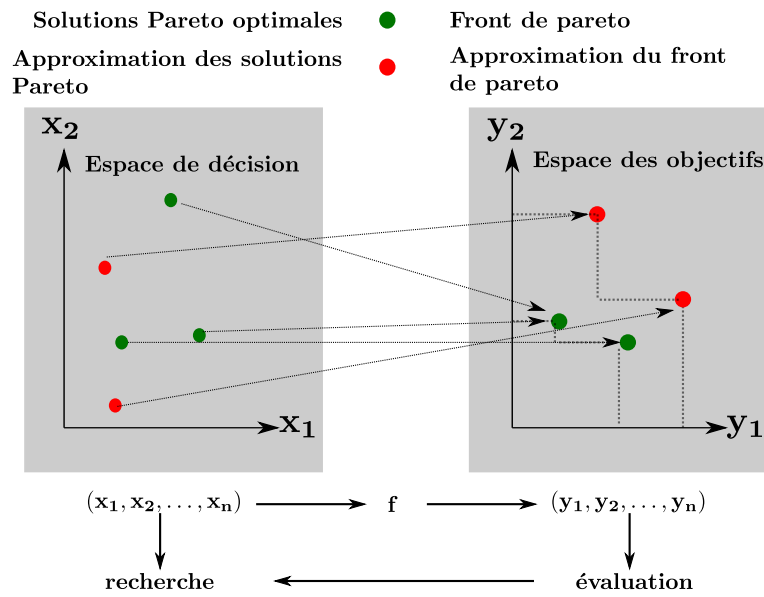


FIGURE 3.3 – Illustration générale d'un POM

**3.3.4 Catégories d'algorithmes**

Le potentiel des algorithmes évolutionnaires pour la résolution des POM a été évoqué pour la première fois à la fin des années 1960 par Rosenberg [Rosenberg 1967]. La première implémentation réelle de ce qu'on appelle maintenant un algorithme évolutionnaire multiobjectif (MOEA) a été l'algorithme VEGA (Vector Evaluation Genetic Algorithm) de Schaffer [Schaffer 1985], principalement pour résoudre les problèmes dans l'apprentissage machine. Depuis lors, une grande variété d'algorithmes ont été proposés dans la littérature. Les MOEAs peuvent se répartir en trois catégories :

- Approches basées sur les fonctions d'agrégation (approche mono-objectif)
- Approches basées sur la population
- Approches Pareto

Nous allons examiner brièvement chacune d'elles dans les sections suivantes.

### 3.4 Transformation du POM en un problème mono-objectif

Il s'agit à présent de présenter deux méthodes assurant la transformation d'un POM en un problème d'optimisation mono-objectif puis de décrire quelques algorithmes d'optimisation mono-objectif.

#### 3.4.1 Méthodes d'agrégation

Les méthodes d'agrégation d'objectifs fusionnent les différentes fonctions objectives pour se ramener à un problème d'optimisation mono-objectif.

Parmi ces méthodes on trouve La méthode de pondération des fonctions objectif [Coello 1998], qui est la plus simple des méthodes d'optimisation multiobjectif. Elle consiste à combiner l'ensemble des objectifs en une seule utilisation soit une addition, multiplication ou toute autre combinaison d'opérations arithmétiques. Un exemple de cette transformation est une somme linéaire de pondérée de la forme :

$$\min \sum_{i=1}^k w_i f_i$$

Avec les coefficients de pondération  $w_i \geq 0$  représentant l'importance relative des  $k$  fonctions objectifs du problème. Il est généralement supposé que :

$$\sum_{i=1}^k w_i = 1$$

L'agrégation des fonctions peut être linéaire (comme dans l'exemple précédent) ou non linéaire. Les deux types de fonctions d'agrégation ont été utilisées avec des algorithmes évolutionnaires dans un certain nombre d'applications, avec un succès relatif. L'agrégation des fonctions a été sous-estimée principalement en raison de la limitation bien connue de l'agrégation des fonctions linéaires (ils ne peuvent pas générer des portions non-convexe du front de Pareto indépendamment de la combinaison de pondérations utilisées [Das 1997]).

Cependant, les fonctions d'agrégation non-linéaires ne présentent pas nécessairement une telle limitation [Coello 2006]. En fait, même les fonctions d'agrégation linéaires peuvent être intelligemment définis de telle sorte que des Fronts de Pareto concaves peuvent être générés [Jin 2001].

#### 3.4.2 Méthode $\varepsilon$ -contrainte

Elle est appelée aussi méthode de compromis [Miettinen 1999]. La transformation d'un POM en un problème mono-objectif se fait de la façon suivante :



1. Choisir un objectif  $i$  à optimiser prioritairement,
2. Choisir un vecteur de contraintes initiales  $\varepsilon = (\varepsilon_1, \dots, \varepsilon_{i-1}, \varepsilon_{i+1}, \dots, \varepsilon_k)$
3. Transformer le problème en gardant l'objectif prioritaire et en transformant les autres objectifs en contraintes d'inégalités comme suit :

$$\text{Minimiser } f(X) = (f_1(X), f_2(X), \dots, f_k(X)), k \geq 2$$

$$\Downarrow$$

$$\begin{aligned} \text{Minimiser } f_i(X) \quad & i \in \{1, \dots, k\} \\ \text{avec } f_j(X) \leq \varepsilon_j \quad & j = \{1, \dots, k\}, j \neq i \end{aligned}$$

(3.6)

Cette approche n'est pas facile à mettre en place à cause de la formulation des préférences des objectifs qui est délicate, nécessitant une connaissance approfondie du problème de départ permettant la priorisation. En plus, les contraintes rajoutées compliquent la résolution du problème.

### 3.4.3 Les algorithmes génétiques

Les algorithmes génétiques (AGs) sont des techniques d'optimisation puissantes, inspirées introduits par ([Holland 1992] et théories darwiniennes) : l'algorithme se charge de maintenir un ensemble de solutions candidates du problème (*individus*), constituant une *population*. Celle-ci s'adapte au cours des générations, par le biais d'opérations successives de sélection, de croisement et de mutation. Chaque individu est évalué et l'ensemble tend à favoriser les individus les mieux évalués dans la population.

En raison de leur robustesse et de leur personnalisation aisée, les AGs ont été appliquées dans la résolution d'un large éventail de problèmes complexes d'optimisation dans l'industrie. Nous citons ci-dessous quelques domaines d'application.

En ingénierie nucléaire, nous pouvons citer les travaux de [Chapot 1999] et de [Sacco 2002] pour la gestion du refroidissement des réacteurs, [Yang 1999] pour l'allocation de fiabilité dans les centrales nucléaires, [Pereira 2003], et [Marseguerra 2004] pour le calibrage/ ajustement des paramètres des réacteurs et enfin [Pereira 2008] pour la conception (design) du cœur du réacteur nucléaire.

Les AGs ont aussi été utilisés pour la résolution des problèmes de transfert de chaleur [Gosselin 2009] et dans des problèmes d'optimisation de production [Pierreval 1997], [Chan 2001], [Kesen 2010], [Santos 2010].

Dans le domaine de la fiabilité des systèmes et de la sûreté de fonctionnement,

les AGs ont été surtout utilisés dans l'optimisation de la fiabilité traitant notamment les problèmes d'optimisation de redondance des composants [Coit 1996b], [Coit 1996a], [Levitin 1998] et les problèmes d'optimisation de la maintenance [Do Van 2012], [Tsai 2001]. Dans le cadre plus précis de l'optimisation des politiques de maintenance préventives nous trouvons dans la littérature beaucoup de travaux qui diffèrent selon le type de la maintenance préventive, considéré ainsi que le contexte et la nature des systèmes à maintenir. Pour l'optimisation de la maintenance préventive systématique, nous trouvons par exemple les travaux de [Lapa 2000], [Tsai 2001] et [Kumar 2012].

### 3.4.3.1 Principe

L'algorithme commence avec un ensemble de solutions possibles du problème (*individus*), constituant une *population*. Les individus sont des vecteurs dont les composants sont des variables, qui sont les *paramètres* à ajuster dans le problème à traiter (par exemple, une longueur, une distance, un temps ... ).

La population est conçue aléatoirement à l'intérieur de l'espace de recherche en respectant les limites prédéfinies (contraintes) par le problème à traiter. Ainsi un individu représente un point de l'espace de décision et devient donc une solution potentielle. Pour mesurer l'adaptabilité de chaque individu au problème, on lui associe une mesure de performance qui correspond à la valeur du critère à optimiser. Pour progresser d'une population d'individus à une autre, on utilise une succession de processus : sélection, croisement et mutation.

- Les solutions qui serviront à former de nouvelles solutions sont aléatoirement sélectionnées par mérite (leurs mesures de performances) : meilleur est l'individu, plus ses chances d'être sélectionné pour se reproduire sont grandes.
- Le croisement et la mutation ont pour but d'assurer l'exploration et l'exploitation de l'espace de recherche.

Ces processus font en sorte qu'au fil des itérations, les meilleurs individus subsistent et participent à la formation des nouvelles générations d'individus alors que les individus les moins adaptés seront souvent éliminés et puis finissent par disparaître. A chaque itération de l'algorithme génétique, une nouvelle génération d'individus est formée. L'algorithme s'arrête lorsque le critère de convergence de l'algorithme est satisfait. Le critère d'arrêt d'un AG peut correspondre, par exemple, à un nombre maximal d'itérations de l'algorithme à ne pas dépasser ou bien à un seuil maximal de performance de la meilleure solution de la population.

L'élaborer de cet algorithme dans sa forme la plus générale nécessite tout d'abord de préciser plusieurs points : le type de codage des solutions, la fonction de mesure de performance, les procédures de reproduction (croisements et mutation) et d'autres paramètres de réglage tel que la dimension de la population, le critère

d'arrêt et les probabilités de croisement et de mutation.

### Sélection

A chaque itération, des individus sont sélectionnés pour se reproduire et pour survivre ou disparaître. Généralement la capacité d'un individu à être sélectionné, dépend de sa performance. Conformément à la théorie darwinienne, les individus les plus performants ont la probabilité la plus élevée de se reproduire dans la population, tandis que ceux qui ont une performance plus faible auront tendance à disparaître.

Il se peut, selon la variante de l'algorithme, que l'opérateur de croisement ou de mutation ne favorise pas les bons individus, ou bien qu'un individu présentant une faible performance, ait certains paramètres optimaux (présents dans la solution optimale). C'est pourquoi il est nécessaire que l'application simultanée des ces deux derniers opérateurs ensemble soit appréhendée avec précaution pour favoriser la sélection des meilleurs individus.

Parmi les méthodes de sélection les plus couramment utilisées dans la littérature, nous pouvons citer la sélection par rang, la sélection par roulette biaisée [Goldberg 1989] et la sélection par tournoi [Mitchell 1996] qui oppose un ensemble d'individus afin d'en sélectionner les meilleurs avec une certaine probabilité.

Une caractéristique importante d'un opérateur de sélection est la pression de sélection exercée sur la population. Cette notion doit être prise en compte d'une manière équilibrée. Une pression sélective trop forte risque de faire converger l'algorithme prématurément vers un minimum local, ce qui revient à faire disparaître toute forme de diversité possible. Cependant avec une pression sélective trop faible, l'algorithme devient une simple recherche aléatoire avec une progression lente de la population.

### Opérateur de croisement

L'opérateur de croisement prend en entrée deux (ou plus) parents et recombine différents éléments issus de ceux-ci afin de générer un ou plusieurs individus. Lorsque les individus croisés sont génétiquement proches (leurs génotypes diffèrent peu), le croisement a une vocation d'exploitation de l'existant. Ce rôle est cependant modifié lorsque les parents sont très différents. Dans ce cas l'opérateur devient un opérateur d'exploration de l'espace décisionnel.

Alors que l'algorithme génétique canonique effectue le croisement par échange simple des moitiés de chromosomes des parents, d'autres types d'opérateurs ont été proposés comme les opérateurs de croisements en plusieurs points, la fusion de solutions, le croisement linéaire avec des poids tirés aléatoirement, etc.

### Opérateur de mutation

Par analogie avec les théories darwiniennes selon lesquelles le génome d'une population évolue ponctuellement par l'intermédiaire de mutations localisées, l'opérateur de mutation d'un algorithme génétique a pour rôle l'exploration de l'espace des solutions du problème considéré par une série de perturbations aléatoires des chromosomes de la population. Sur une représentation telle qu'une chaîne de bits, cette modification consiste, par exemple, à inverser aléatoirement un ou plusieurs bits. Le fonctionnement de l'opérateur de mutation est stochastique : une mutation peut survenir en chaque gène ou ponctuellement en un gène aléatoirement choisi d'un individu avec une probabilité  $P_{mut}$ .

### **Critère d'arrêt**

On peut décider de l'arrêt des calculs après un certain délai (temps ou nombre d'itérations), après que le meilleur individu ait atteint un certain niveau de qualité ou après un certain nombre d'itérations sans amélioration du meilleur individu, ou bien lorsque l'écart type de la fitness de la population passe en deçà d'un certain seuil.

#### **3.4.3.2 Les paramètres d'un algorithme génétique**

En dehors des différents opérateurs et stratégies entrant en compte dans l'implémentation d'un algorithme génétique, il existe un certain nombre de paramètres à définir :

##### **Taille de la population**

Si de manière intuitive, une population de grande taille paraît être une solution idéale et ce afin de mieux explorer l'espace de recherche, l'augmentation en terme de coût de calculs supplémentaires n'est pas à négliger.

En l'absence d'une stratégie visant à adapter ce paramètre au long du fonctionnement de l'algorithme, le retour d'expérience reste la manière la plus répandue de fixer ce paramètre.

##### **Probabilité de croisement**

Une fois qu'un ensemble d'individus a été sélectionné pour participer à la reproduction, ceux-ci n'y participeront réellement qu'avec une probabilité dite de croisement,  $P_{cross}$ .

##### **Probabilité de mutation**

Sujet de nombreuses discussions, la probabilité de mutation  $P_{mute}$  représente la probabilité avec laquelle un individu (ou un de ses gènes) subira l'effet de l'opérateur de mutation. Si une probabilité élevée permet une grande couverture de l'espace de recherche, la convergence à proximité de l'optimal recherché sera très

difficile.

Inversement, une probabilité trop faible restreint la recherche et diminue d'autant les chances d'approcher de l'optimal ; le paramétrage d'un algorithme génétique est donc délicat. Souvent, le paramétrage est établi soit à partir d'une connaissance a priori du problème à résoudre, soit, le plus souvent, à partir d'un plan d'expérience et de nombreux essais. Nous verrons que certaines recherches ont visé récemment à automatiser le paramétrage en lui faisant suivre ce dernier, au même titre que la population elle-même, un processus évolutionnaire.

Il faut rappeler qu'il n'existe pas de paramétrage universellement performant pour un algorithme génétique. S'il existe des valeurs acceptées pour certains paramètres on emploie souvent une probabilité de croisement proche des 0,80 ainsi qu'une probabilité de mutation en  $\mathcal{O}(\frac{1}{n})$  avec  $n$  le nombre de gènes du chromosome, il est presque toujours nécessaire de passer par une phase d'étalonnage.

#### 3.4.4 La recherche tabou

La recherche avec tabous (Tabu Search) est une méthode mise au point par Glover [Glover 1986]. La particularité de la méthode réside dans l'usage d'une mémoire. Le principe de la méthode est simple :

L'algorithme part d'une solution quelconque. Puis, en perturbant cette solution, on génère  $n$  voisins. Ensuite, on examine les  $n$  voisins et on retient pour solution courante à l'itération suivante le meilleur des  $n$  voisins, même s'il est moins bon que la solution précédente. Ainsi, on évite le blocage de l'algorithme dans un minimum local.

Toutefois, pour éviter le risque de bouclage (retourner à une solution précédemment visitée) qui augmente lorsqu'on accepte une solution plus mauvaise que la solution courante, on interdit provisoirement toutes modifications qui renvoient à une solution déjà visitée.

Cette stratégie d'interdiction provisoire est élaborée grâce à une liste "Tabou"  $T$ , qui contient les solutions déjà visitées. Ainsi, après avoir généré un nouveau voisin de la solution courante, l'algorithme examine si celui-ci n'appartient pas à la liste Tabou. En cas d'appartenance, un autre point sera généré pour le remplacer [Berthiau 2001].

#### 3.4.5 Le recuit simulé

Le recuit simulé a été proposé par [Kirkpatrick 1983] pour la résolution des problèmes d'optimisation combinatoire comme par exemple le problème de déplacement du voyageur de commerce (traveling salesman problem) [Randelman 1986].

La méthode trouve ses origines en thermodynamique. Elle s'inspire du phénomène physique de refroidissement lent d'un corps en fusion qui le conduit à un état solide de basse énergie.

En fait, un corps chauffé à une température très élevée devient liquide et peut occuper toute configuration. Quand la température décroît, le corps va se figer peu à peu dans une configuration qu'il est de plus en plus difficile à déformer. Toutefois, en le réchauffant à nouveau, le corps peut être retravaillé afin de lui donner la forme désirée.

Il faut alors, abaisser lentement la température en marquant des paliers suffisamment longs pour que le corps atteigne l'équilibre thermodynamique à chaque palier de la température, ce qui permet d'obtenir à la fin du processus un corps dans un état cristallin bien ordonné correspondant à un état d'énergie minimum.

Par contre, si la baisse de température se fait de manière trop brutale, le corps devient amorphe et ses atomes sont figés dans un état désordonné traduisant un minimum local d'énergie.

De manière analogue, une procédure de recuit simulé « refroidit » un problème donné en minimisant son état énergétique (i.e. le critère à optimiser). Toutefois, contrairement aux méthodes déterministes de descente, des accroissements de la fonction objectif peuvent survenir en cours de recherche avec une probabilité décroissante lors des itérations.

Cette particularité permet, surtout lors des premières itérations de l'algorithme, de s'extraire des pièges constitués par des optimaux locaux, et compte-tenu de la décroissance de la probabilité d'acceptation de solutions dégradantes, ce dernier a tendance, en fin de recherche, à se comporter comme une classique méthode de descente. Le principe de la procédure est détaillé dans [Cerny 1985] et [Dréo 2005].

L'algorithme du recuit simulé a été également adapté pour l'optimisation multiobjectif. Les adaptations les plus populaires sont la méthode MOSA *Multiple Objective Simulated Annealing* proposée par [Ulungu 1999] et la méthode PASA *Pareto Archived Simulated Annealing* proposée dans [Engrand 1997]. Ces deux méthodes reposent sur un formalisme agrégatif de la fonction objectif. MOSA exploite les propriétés du recuit simulé pour approcher l'ensemble de solutions optimales d'un problème d'optimisation combinatoire multiobjectif. PASA utilise une fonction d'agrégation des objectifs, conjuguée avec une procédure d'archivage des solutions non dominées

### 3.4.6 Algorithmes basés sur l'intelligence collective : Les colonies de fourmis

Les algorithmes fondés sur l'intelligence collective, proposés par [Coloni 1991] utilisent une analogie avec le comportement naturel d'une colonie d'insectes. En observant une colonie de fourmis à la recherche de nourriture dans les environs du nid, on s'aperçoit qu'elle résout des problèmes tels que celui de la recherche du plus court chemin. Les fourmis résolvent des problèmes complexes par des mécanismes assez simples à modéliser.

En optimisation mono-objectif, les algorithmes de colonies de fourmis sont très appréciés pour résoudre des problèmes combinatoires. En revanche, peu de travaux existent dans le cas multiobjectif : quelques références à des algorithmes multiobjectif sont citées dans ce travail de [Blum 2005].

Le fonctionnement de l'algorithme s'apparente à des fourmis recherchant leur nourriture (i.e. la solution au problème proposé). Comme les AGs, ces algorithmes s'appuient sur une population d'individus.

La recherche s'effectue en deux phases :

1. Recherche locale : les fourmis ayant la meilleure valeur du critère à optimiser effectuent une recherche locale autour de leur position.
2. Recherche globale : les autres fourmis effectuent une recherche plus globale en suivant les phéromones déposées antérieurement. Cette phase de suivi de phéromone comporte une procédure d'évaporation permettant de ne retenir, au fur et à mesure de l'exploration, que les traces les plus performantes.

## 3.5 Approches non Pareto

Dans ce type de métaheuristiques, la population de l'Algorithme Evolutionnaire (AE) est utilisée pour diversifier la recherche.

### 3.5.1 Algorithme VEGA

En 1985 [Schaffer 1985] propose une extension d'un algorithme génétique simple pour la résolution d'un POM. Cette méthode est appelée *Vector Evaluated Genetic Algorithm* (VEGA). VEGA se compose essentiellement d'un algorithme génétique simple avec un mécanisme de sélection modifié. À chaque génération, un certain nombre de sous-populations sont générées en effectuant une sélection propre à chaque fonction objectif.

Ainsi, si nous avons  $k$  objectifs et une population de  $n$  individus, une sélection de  $n/k$  individus est effectuée pour chaque objectif. Ainsi  $k$  sous-populations vont



être créées, chacune d'entre elles contenant les  $n/k$  meilleurs individus pour un objectif particulier. Ces sous-populations sont ensuite mélangées ensemble pour obtenir une nouvelle population de taille  $n$ , sur laquelle l'algorithme génétique (GA) applique les opérateurs de croisement et de mutation.

VEGA propose plusieurs problèmes, dont le plus important est que ses schémas de sélection sont opposés à la notion de dominance au sens de Pareto. En effet, dans l'algorithme VEGA, la méthode de sélection ne tient compte que d'un seul objectif favorisant la création de sous-populations dont les meilleurs individus sont spécialisés pour un objectif particulier. Ainsi, les individus ayant une performance générale acceptable et qui ne figurent pas parmi les meilleurs pour aucun objectif (appelé par Schaffer les individus "milieu") vont être éliminés car ils ne seront sélectionnés dans aucune sous-population. Ceci empêche la méthode d'atteindre les solutions "compromis".

Pour faire face à ce problème, quelques heuristiques ont été proposées. Schaffer proposait d'utiliser une méthode de sélection par préférence pour les individus non dominés, ou des méthodes de croisement avec sélection non aléatoire.

### 3.5.2 Utilisation des genres

Dans [Allenson 1992], les auteurs proposent une méthode qui utilise la notion de genre et d'"attracteur sexuel" pour traiter un problème à deux objectifs. Cet algorithme a été appliqué dans un problème de planification d'itinéraire de pipeline : minimiser la longueur d'un pipeline tout en réduisant l'impact destructeur, causé par sa construction, sur l'environnement. En utilisant deux genres, l'algorithme est a été étendu pour prendre en compte à la fois, le critère économique et environnementale, dans seul plan de reproduction.

## 3.6 Approches Pareto

Les métaheuristiques les plus récentes et les plus représentatives de cette approche sont les suivantes.

### 3.6.1 NSGA-II

[Deb 2000] ont proposé une version révisée plus efficace de la NSGA (*Non-dominated Sorting Genetic Algorithm*) [Srinivas 1994], appelé NSGAI. Cette version n'utilise pas de mémoire externe. Son mécanisme élitiste consiste à combiner les meilleurs parents avec les meilleurs descendants obtenus. A chaque nouvelle génération, les meilleurs individus rencontrés sont conservés.



L'opérateur de sélection dans NSGA-II est basé sur l'élitisme (dominance au sens Pareto) et sur le calcul de la distance de surpeuplement (*crowding*) de chaque individu, pour maintenir la diversité de la population.

L'algorithme commence par créer une population d'enfants  $Q_t$  en utilisant une population de parents  $P_t$ . Les deux populations réunies forment une population combinée  $R_t$  de taille  $2N$ .

$R_t$  est ensuite triée en rangs de Pareto pour former les fronts successifs. Cette procédure de *ranking* se déroule comme suit :

Le premier front  $F_1$  correspond à l'ensemble des solutions non-dominées de  $R_t$ . Le deuxième front  $F_2$  est constitué des individus non-dominés de l'ensemble  $(R_t \setminus F_1)$ . Cette procédure est répétée jusqu'à ce que tous les individus de  $R_t$  soient attribués à un front.

La nouvelle population  $P_{t+1}$  est créée et est remplie au fur et à mesure avec les fronts successifs. Comme la taille de la population  $R_t$  est  $2N$ , les fronts successifs ne peuvent pas alors intégrer en totalité  $P_{t+1}$  qui est de taille  $N$ . Ces fronts seront tout simplement éliminés.

Cependant, il se peut que la taille du dernier front considéré soit supérieure aux nombres de places vacantes à remplir dans la nouvelle population. Dans ce cas, le critère de surpeuplement décrit dans la section précédente sera utilisé pour choisir parmi les solutions du dernier front, celles qui vont intégrer la nouvelle population afin de favoriser les régions les moins peuplées du front considéré (diversité des solutions).

### 3.6.2 SPEA2

L'algorithme SPEA2 (*Strength Pareto Evolutionary Algorithm 2*), proposé par [Zitzler 2002] a été développé comme une amélioration de SPEA [Zitzler 1999a].

SPEA est un algorithme élitiste qui utilise une population externe, appelée archive, qui contient les meilleurs solutions de compromis rencontré durant l'exécution de l'algorithme. La population d'archive doit contenir un nombre limité de solutions non-dominées.

A chaque itération de l'algorithme, les solutions de l'archive sont comparées, au sens de dominance, à des nouvelles solutions. Seules, les solutions non dominées sont conservées dans l'archive.

Si la taille de population d'archive dépasse une certaine limite (fixée a priori), elle sera soumise à une procédure de clustering. Il est à noter, également, que dans SPEA2, seuls les membres de l'archive participent au processus de reproduction. Par comparaison avec SPEA, SPEA2 diffère en termes de gestion de l'archive et

de densité des solutions :

- SPEA2 repose sur une archive notée  $A_t$  de taille fixe  $N_A$ . A chaque itération, les nouveaux individus non-dominés de la population  $P_t$  sont comparés aux ceux de l'archive  $A_t$  en utilisant le critère de dominance. Si le nombre d'individus non-dominés n'est pas suffisant, l'archive est complétée par les meilleurs individus dominés. Il est à noter qu'une telle situation ne peut se produire que durant les premières générations.
- Comparé à SPEA, le calcul de performance dans SPEA2 tient mieux en compte la densité des solutions qui représente un critère de diversité. Le calcul de la fitness d'un individu  $i$  se fait comme suit :

$$f(i) = R(i) + S(i)$$

où,  $R(i)$  est la performance préliminaire (*raw fitness*) de l'individu et  $D(i)$  est sa densité calculée par la technique des kppv.  $R(i)$  est calculée sur la base des valeurs de "Strength" des solutions.  $R(i) = 0$  correspond à un individu non dominé, tandis qu'une valeur élevée de  $R(i)$  correspond à un individu dominé par plusieurs individus. La limite de ce critère réside dans le cas où plusieurs individus ne se dominent pas les uns les autres.

### 3.6.3 $\epsilon$ -MOEA

$\epsilon$ -MOEA [Deb 2003] est un algorithme évolutionnaire multiobjectif. Il utilise un schéma d'évolution basé sur le concept de l' $\epsilon$ -dominance introduit dans [Laumanns 2002].

Ce concept sera expliqué plus tard dans 4.10.2. L'idée principale de l'algorithme est de subdiviser l'espace de recherche en un certain nombre de divisions puis de s'assurer que chaque division ne contient qu'une seule solution au plus.

$\epsilon$ -MOEA considère deux populations évoluant en même temps : une population principale de l'algorithme évolutionnaire notée  $P(t)$  et une population archive notée  $A(t)$  (où  $t$  représente le compteur des générations).

A chaque itération de l'algorithme,  $k$  solutions  $\lambda$  sont créées en combinant à chaque fois une solution  $i \in P(t)$  et une autre  $j \in A(t)$ . Par la suite les archives  $A(t)$  et  $P(t)$  sont mises à jour avec les  $k$  solutions créées. Pour intégrer  $A(t)$ , une solution  $\lambda$  doit  $\epsilon$ -dominer toutes les solutions de  $A(t)$ . Cependant, pour être admise dans  $P(t)$ ,  $\lambda$  doit dominer au sens Pareto au moins une ou plusieurs solutions de la population. Elle remplacera alors aléatoirement une de ses solutions dominées.

### 3.6.4 MO-CMA-ES

L'algorithme MO-CMA-ES (*MultiObjective-Covariance Matrix Adaptation Evolution Strategy*) [Igel 2007] est une extension de l'algorithme mono-objectif CMA-ES [Hansen 2001]. Il s'agit d'une adaptation des stratégies d'évolution de l'algorithme CMA-ES au contexte multi-objectif : la matrice de co-variance et les pas de transition sont adaptés, au cours des itérations, aux caractéristiques du paysage de la fitness locale.

Plusieurs stratégies d'évolution comme la variante  $\mu \times (1 + 1)$  peuvent être utilisées dans MO-CMA-ES. Initialement, une population est générée aléatoirement et puis évaluée. Ensuite, à chaque itération  $t$ ,  $\mu$  parents, génèrent chacun  $\lambda = 1$  enfant. Après l'évaluation d'un enfant généré, les parents et un enfant sont unis pour former la population  $Q_t$ .

Si les mutations ont permis d'avoir un descendant meilleur que le parent, les paramètres du processus de mutation du parent et de son enfant, sont alors modifiés. Par la suite, la matrice de covariance de l'enfant est également réactualisée en considérant la mutation précédemment effectuée.

Après,  $Q_t$  sera triée selon le principe de dominance et les  $\mu$  meilleurs individus seront sélectionnés pour former la population des parents de la génération suivante.

Si, le principe de dominance ne permet pas de départager les individus non-dominés entre eux, MO-CMA-ES, en plus un autre critère de comparaison. Selon la nature de ce deuxième critère de comparaison, MO-CMA-ES se décline en deux variantes : c-MO-CMA-ES, utilisant le critère de surpeuplement du NSGA-II et s-MO-CMA-ES qui se sert de la mesure de contribution à l'hypervolume.

## 3.7 Conclusion

Dans ce chapitre un état de l'art de l'optimisation évolutionnaire multiobjectif a été présenté. Après avoir défini le problème d'optimisation multiobjectif et ses éléments de base relatifs nous avons passé en revue quelques métaheuristiques en décrivant le principe de fonctionnement de chacun.

Nous consacrons le prochain chapitre à la description des métaheuristiques que nous avons développés pour la résolution du problème de l'optimisation de la maintenance préventive.



# Adaptation des Métaheuristiques pour l'optimisation de la maintenance préventive

---

## Sommaire

4.1 Introduction . . . . .	87
4.2 Généralités sur l'approche . . . . .	88
4.3 Codage de la solution candidate . . . . .	89
4.4 Initialisation des solutions . . . . .	90
4.5 Mesure de qualité de la solution candidate . . . . .	91
4.6 Choix de la taille de la population . . . . .	92
4.7 Algorithme génétique pour l'optimisation de la maintenance préventive périodique AG-MP(P) . . . . .	92
4.8 Algorithme génétique pour l'optimisation de la maintenance préventive non périodique AG-MP(NP) . . . . .	97
4.9 Recuit simulé modifié (RSM) pour l'optimisation de la main- tenance préventive . . . . .	113
4.10 L'algorithme ev-MMOGA pour l'optimisation de la mainte- nance préventive . . . . .	118
4.11 NSGA-II modifié pour l'optimisation de la maintenance pré- ventive périodique . . . . .	129
4.12 Conclusion . . . . .	131

---

## 4.1 Introduction

Comme introduit dans les chapitres précédents, un modèle de l'évaluation de l'état d'un système multi-composants, multi-états, est nécessaire pour comprendre son comportement sur sa durée de vie.

Pour une modélisation complète du système, nous avons vu qu'il convient également de rajouter une modélisation de la maintenance incluant les systèmes de surveillance (diagnostic) et les actions (réparation, remplacement ...). Ainsi, nous sommes en capacité de réaliser une évaluation de la politique de maintenance. Pour ce faire, plusieurs objectifs ou critères sont à définir (mesure d'évaluation) et leurs minimisation simultanée est le problème d'optimisation multiobjectif que l'on se pose. Cette optimisation s'effectue en agissant sur les paramètres de réglage de la politique de maintenance inclus dans le modèle.

Dans ce chapitre, nous nous plaçons dans le cadre de l'optimisation d'une stratégie de maintenance préventive systématique à l'aide d'algorithmes évolutionnaires. Des algorithmes évolutionnaires, mono-objectifs et multiobjectif ont été alors élaborés pour la résolution de ce problème. Dans ces algorithmes, plusieurs points sont à définir pour lesquels différents choix sont possibles. Nous présentons d'abord des généralités sur l'approche, ensuite nous détaillerons chacun de ces algorithmes.

## 4.2 Généralités sur l'approche

L'objectif principal de réglage d'une stratégie de maintenance préventives systématique est d'optimiser les intervalles de temps entre les interventions de maintenance planifiées pour chaque composant du système afin de minimiser son coût de fonctionnement, le coût de maintenance du système, et de garder une bonne disponibilité. Quand les intervalles entre deux opérations de maintenance, d'un composant donné sont réguliers, on parle de MP périodique notée MPP. Dans le cas contraire, il s'agit de MP Non Périodique qu'on notera MPNP.

De manière générale la modélisation d'un problème d'optimisation par un algorithme évolutionnaire se doit de préciser préalablement les points suivants :

- (i) définition de l'un individu dont la structure de données code la solution candidate ;
- (ii) Définition d'une fonction objectif qui évalue la solution candidate ;
- (iii) Ajustement des paramètres de réglage : taille de la population, taux de croisement et de mutation, critères d'arrêt.

## 4.3 Codage de la solution candidate

Le choix du type de codage des individus est primordial dans la construction d'un algorithme évolutionnaire. Plusieurs types de codage sont possibles dans

la littérature : le codage binaire, le codage de Gray et le codage réel. Le codage binaire  $\{0,1\}$  [Chipperfield 1994] est le codage le plus populaire. Dans ce type de codage, les solutions sont codées sous la forme d'une suite particulière de 0 et 1 de longueur fixe, s'adaptant ainsi facilement au codage de différents types d'éléments (réels, entiers, booléens) [Mitchell 1996]. D'ailleurs, La plupart des opérateurs de reproduction ont été développées autour de ce type de codage. La représentation binaire d'une solution est désignée par un génotype. La valeur réelle de ce dernier dans l'espace de recherche est appelée phénotype.

Dans le cadre de notre étude, nous traitons l'optimisation de la maintenance des systèmes multi-composants. Cela implique que l'individu doit essentiellement coder le calendrier de maintenance de chaque composant du système. Deux représentations sont alors proposées : une représentation binaire et une représentation réelle.

Pour la représentation binaire (génotype), la solution candidate doit encoder tout le calendrier des interventions de maintenance planifiées pour l'ensemble des composants du système depuis le début de la mise en fonctionnent du système jusqu'à son remplacement total. Le génotype se présente sous une forme ma-

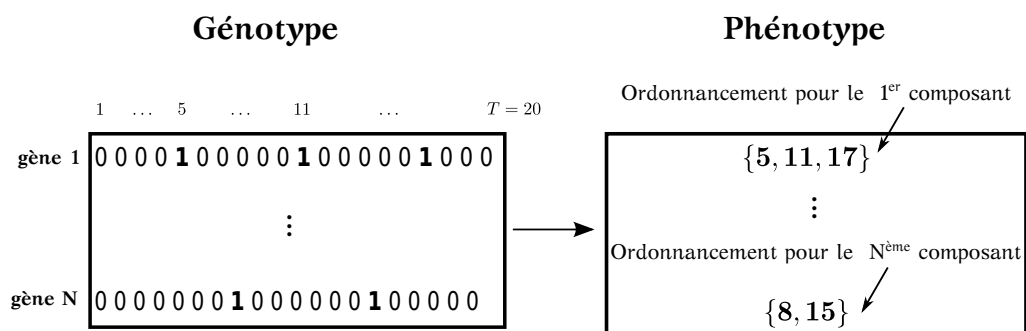


FIGURE 4.1 – Représentation binaire d'un individu

tricielle de dimension  $N \times T$ .  $N$  correspond au nombre de composants dans le système et  $T$  correspond à l'horizon temporelle de l'étude. Le décodage d'un gène (ligne de la matrice du génotype) est tel qu'un bit égal à 1 signifie qu'une action de maintenance est prévue à cette date. La figure 4.1 montre le génotype et son décodage pour chaque composant, représenté par un vecteur dont les éléments sont les dates d'intervention dans une échelle temporelle  $T$ . Ce codage permet la représentation d'une maintenance préventive non périodique ou périodique.

Cependant, il n'est pas avantageux, d'utiliser la représentation binaire pour coder un calendrier de maintenance préventive périodique (MPP), alors qu'il suffisait de représenter seulement les périodes (fréquences) de maintenance de chaque com-

posant pour déduire par la suite les instants de maintenance. Ainsi, pour coder une solution décrivant une politique de MPP, nous employons un codage de type réel 4.2.

**Individu**

10	25	6		8		12
$\theta_1$	$\theta_2$	$\theta_3$		$\theta_i$		$\theta_N$

FIGURE 4.2 – Phénotype d'un individu pour une MPP. Les dates d'interventions sur le composant  $i$  sont  $\{8, 16, 24, 32 \dots\}$

Les individus se présentent la forme de vecteurs à valeurs réelles. La longueur de chaque vecteur est égale au nombre de composants du système. Dans ce sens, chaque variable du vecteur correspond à une période de maintenance pour un composant donné du système.

## 4.4 Initialisation des solutions

L'étape de création et d'initialisation de population initiale représente une étape crucial dans le processus de construction de l'AE. La population initiale, représente le point de départ des générations futures de solutions. Elle influence donc, fortement, la qualité des populations futures et par conséquent la vitesse et l'efficacité de l'AE en général.

Nous avons tendance à penser que l'algorithme trouvera rapidement des optimums si la population initiale contient déjà des solutions quasi-optimales. Cependant, ces solutions quasi-optimales dont la performance est très supérieure au reste des individus de la population peuvent, à terme, piéger l'algorithme dans des optimums locaux.

En effet, les meilleurs individus dominants sont beaucoup plus sollicités pour engendrer les nouvelles solutions (enfants), et puis de génération en génération la population s'appauvrit en contenant des individus de plus en plus similaires. Inversement, une très grande diversité d'individus (performances équivalentes) mènerait à une recherche très aléatoire. Il est donc essentiel de trouver une technique de génération de la population initiale qui établit un équilibre entre la qualité et la diversité des solutions. Trois approches de génération de la population initiale peuvent être identifiées :

- Une construction totalement aléatoire de solutions : elle est souvent utilisée si



la position de l'optimum dans l'espace de recherche est totalement inconnue. Les individus sont générés aléatoirement tout en respectant les contraintes du problème d'optimisation.

- Une construction à l'aide d'heuristiques : si nous disposons d'informations à priori sur le problème ou sur la localisation de l'optimum, cette technique de génération pourrait être utilisée. Les individus seront générés dans un sous-espace particulier de l'espace de recherche, afin d'accélérer la convergence.
- une construction mixte à l'aide d'une procédure réunissant les deux principes précédents : cette technique est utilisée, si nous disposons d'informations sur le problème, mais pas assez pour pouvoir utiliser l'approche de génération à l'aide d'heuristiques.

Dans les algorithmes proposés dans notre travail, nous avons employé ces approches seules ou combinées pour générer les individus de la population initiale et parfois pour la création de nouvelles solutions injectées à la population au cours des générations. Le processus de sélection de candidats (parents) pour générer les solutions (enfants) à l'aide d'opérateurs génétiques de reproduction, doit fonctionner correctement et permettre d'améliorer la qualité des solutions tout en préservant la diversité au cours des générations.

## 4.5 Mesure de qualité de la solution candidate

L'évaluation de la qualité de chaque individu (solution potentielle) est réalisée à chaque fois par le calcul du coût total du fonctionnement du système avec les nouveaux paramètres de l'individu. Dans ce travail, l'utilité moyenne *fitness* (appelée aussi, force, adaptation ou encore fonction d'évaluation) de l'état du système exposé en section 2.6.4.2 représente à la fois le critère à optimiser et la fonction d'évaluation des individus. Chaque fois qu'un nouvel individu est créé, une valeur qui mesure la performance de la solution lui est forcément associée. Cette valeur intervient dans le critère de sélection d'un individu par rapport à un autre pendant les processus de sélection. De ce fait, la fonction d'évaluation de performance doit être en concordance avec la fonction objectif du problème à résoudre.

Dans le cas d'un problème mono-objectif, la fonction de performance reproduit la fonction objectif ou une de ses sous fonctions. Toutefois, dans le cas multiobjectif, la fonction d'évaluation peut inclure, également, des facteurs en relation avec les contraintes et objectifs du problème, comme par exemple une pénalité sur les contraintes non satisfaites.

## 4.6 Choix de la taille de la population

La taille de la population, est un paramètre de réglage de l'algorithme, souvent déterminé de façon empirique. Elle peut être variable au cours de l'exécution et doit être en concordance avec la longueur des génotypes et les objectifs du problème d'optimisation.

Selon certains auteurs, il est important de considérer au début de l'algorithme une grande taille de population [De Jong 2007], puisque l'on traite typiquement un espace complexe, potentiellement mal conditionnée.

L'espace de recherche pour un problème d'optimisation de la maintenance, varie selon le contexte et la nature du système à analyser ainsi que le type de la stratégie de maintenance à optimiser. Cet espace est d'autant plus grand que le système se décompose en de nombreux sous-systèmes et composants. En tenant compte de ces différents éléments, nous pouvons fixer la taille de la population initiale. Celle-ci reste invariante au fil des générations de l'AE, à moins qu'on spécifie le contraire dans l'algorithme.

## 4.7 Algorithme génétique pour l'optimisation de la maintenance préventive périodique AG-MP(P)

### 4.7.1 Initialisation des solutions et limites de l'espace de recherche

Réduire l'espace des paramètres revient en quelques sortes à minimiser le coût de calcul global. En effet, en diminuant la complexité de la recherche, le nombre de générations nécessaires pour converger vers l'optimum global peut être réduit.

En se basant sur certaines spécifications et contraintes fonctionnelles, techniques, matérielles et financières (budget, seuil sécuritaire, ...) une estimation des valeurs maximales et minimales de la fonction objectif peut être connues. On peut alors stipuler un niveau maximum  $\text{Seuil}_{\max}$  (ou minimum  $\text{Seuil}_{\min}$ ) pour la fonction objective. Ainsi, les solutions ne vérifiant pas ces spécifications sont éliminées (non acceptées dans la population) et les limites de recherche seront redéfinies.

Toutefois, ces vérifications, par évaluation et comparaison de l'acceptabilité des solutions candidates, coûtent aussi cher en complexité calculatoire. L'idée serait de pouvoir écarter ces solutions avant même de les évaluer. Disposant des niveaux  $\text{Seuil}_{\max}$  (ou  $\text{Seuil}_{\min}$ ), on pourrait alors, par une projection plus au moins simple selon la dimension de la fonction objectif, redéfinir un ou plusieurs sous-espaces de variables correspondantes aux variables du vecteur de solution

acceptables. Cette opération dépend certainement de l'expérience de l'utilisateur.

La réduction de l'espace de recherche des variables conçues augmente la précision du résultat final et facilite la recherche dans la direction de l'optimum globale en un temps plus concis. La Figure 4.3 illustre cette procédure dans un problème

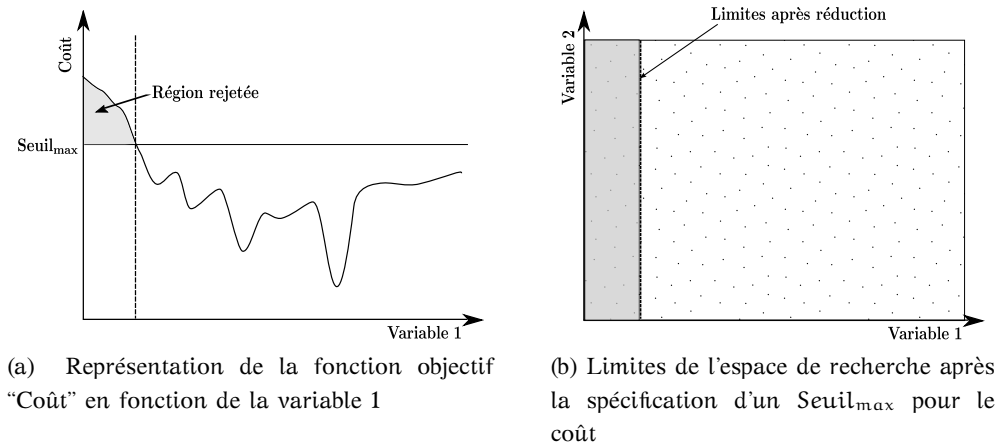


FIGURE 4.3 – Exemple simple de réduction de l'espace de recherche

mono-objectif fictif de minimisation de coût. Après la spécification d'une valeur maximale pour le coût, on peut évincer une bonne partie de l'espace de variables. Pour mieux explorer la région restreinte, une population est créée à l'intérieur de ces limites redéfinies.

Dans certains cas la réduction de l'espace telle qu'elle est expliquée ci-dessus est difficile à mettre en œuvre en raison du manque des connaissances sur le comportement de la fonction objectif ou aussi à cause du nombre élevé des variables à gérer dont plusieurs sont corrélées.

#### 4.7.2 Déroulement de l'algorithme

##### Données :

- Les paramètres de l'algorithme génétique
- Modèle d'évaluation de la politique de maintenance

**Résultat : un calendrier de maintenance préventive périodique quasi-optimal**

---

**Algorithme 2** : Algorithme Génétique pour l'optimisation de la MPP

---

```

t = 0 ;
% Génération de la population initiale
Pt.sol = GenerPop (Pt.size);
% Evaluation des individus Pt.sol de la population Pt
Pt.fit = EvalPop(Pt.sol);
Tri(Pt);

% Pt.nbgen : nombre maximal de générations
tant que (t < Pt.nbgen) faire
    Ptsel = Selection (Pt);
    Ptcro = Croisement (Ptsel) ;
    Ptmut = Mutation (Ptcro) ;
    % Formation de la nouvelle population
    Pt+1 = Combinaison (Pt, Ptmut) ;
    % Puissance de perturbation de la population
    nt = PuissancePerturb (Pt+1);

    si nt > 0 alors
        % Génération de la population de perturbation
        Ptper.sol = GenerPop (nt);
        Ptper.fit = EvalPop (Ptper.sol);
        % injecter nt nouveaux individus dans Pt+1
        Pt+1 = Combinaison (Pt, Ptmut) ;
        Tri (Pt+1);
    fin
    t = t + 1
fin

```

---

### 4.7.3 Sélection d'individus

Dans cette étape et à chaque itération de l'algorithme, des individus sont sélectionnés pour se reproduire, survivre ou disparaître. Généralement, la capacité d'un individu à être sélectionné dépend de sa performance. Cependant, nos premiers essais ont montré que l'emploi d'une stratégie de sélection élitiste<sup>1</sup>, ou bien d'une sélection par roulette (la probabilité qu'un individu soit sélectionné pour

---

1. N/2-élitisme : les individus sont triés selon leur mesure de qualité. Seule la moitié supérieure de la population, correspondant aux meilleurs composants, est sélectionnée.

la reproduction est directement proportionnelle à sa qualité), débouchait sur une convergence prématurée de l'algorithme. Nous optons alors pour une méthode offrant une meilleure exploration de l'espace de recherche : une sélection par tournoi.

Cette stratégie permet de favoriser les individus les mieux adaptés, tout en laissant aux individus les plus faibles l'opportunité de participer au processus d'évolution.

Soit une population comportant  $\lambda$  individus. A la suite de l'étape de sélection, les individus sélectionnés ont une probabilité  $P_{\text{cross}}$  de se reproduire.

Une fois la phase de croisement terminée, l'opérateur de mutation est appliqué aux  $\gamma$  individus issus de la phase de croisement. A chaque itération, la population  $\text{Pop}_{t+1}$  est composée des  $\lambda - \gamma$  meilleurs individus de  $\text{Pop}_t$  et des  $\gamma$  individus venant d'être générés. La stratégie élitiste, à ce niveau, garantit la convergence de l'algorithme. Il est cependant préférable de ne conserver qu'un nombre réduit d'individus d'une génération à l'autre, afin d'éviter une convergence trop rapide.

#### 4.7.4 Opérateurs de reproduction

Dans les AGs, le croisement et la mutation sont considérés comme les principaux opérateurs pour produire de nouveaux individus. Un opérateur de croisement peut utiliser  $k$ ,  $k > 1$  individus (parents) pour former un ou plusieurs nouveaux individus (descendants), en leur transférant certains paramètres des individus parents. Ces opérateurs favorisent ainsi la diversification de la texture de la population, afin de mieux explorer l'espace de recherche. L'utilisation de variables entières dans les problèmes d'optimisation nécessite des opérateurs adaptés. Dans l'AG-MP(P), il a été choisi d'adapter les opérateurs de croisement et de mutation des variables continues.

Pour le croisement, nous utilisons la méthode de croisement BLX- $\alpha$  volumique [Eshelman 1992] suivie d'un changement d'ensemble. Soient  $X$  et  $Y$  deux individus choisis pour être mutuellement croisés.  $x_i$  et  $y_i$  sont les composantes respectives de  $X$  et  $Y$ , tel que  $1 \leq i \leq n$  et  $n$  est la dimension de l'espace des individus. Soit  $Z$  un descendant du couple  $X$  et  $Y$  tel que :

$$z_i = x_i - \alpha(y_i - x_i) + (1 + 2\alpha)(y_i - x_i)\mathcal{U}(0,1) \quad (4.1)$$

$\mathcal{U}(0,1)$  désigne un nombre aléatoire tiré uniformément dans l'intervalle  $[0,1]$ .

Les composantes  $z_i$  du vecteur fils "Z" sont obtenues à partir de la combinaison des  $x_i$  et  $y_i$  comme montré dans 4.1. Par ailleurs, la conversion des  $z_i$  en valeurs de type entier est indispensable afin que  $Z$  soit considéré comme solution faisable et puisse prétendre intégrer la population à la prochaine génération.

Contrairement à l'opérateur de croisement qui peut faire varier significativement la population, la mutation se contente de maintenir un minimum de diversité. Ce dernier a pour mission d'opérer une modification aléatoire sur un individu pour en former un autre qui le remplacera. La proportion des individus mutés dans la population est égale à un taux de mutation défini préalablement dans l'algorithme. Un individu a une probabilité  $P_{mut}$  d'être muté.

La mutation proposée pour l'AG-MP(P) est adaptative. Elle s'appuie sur le principe de la mutation gaussienne avec un changement dynamique de paramètres. Le principe est le suivant : l'opérateur de mutation consiste à ajouter un bruit gaussien  $\Delta$  à un élément  $x_i$ , choisi aléatoirement, d'un individu  $X$  tel que :

$$\Delta = \frac{\mathcal{N}(x_i, 0, \eta t)}{P_t \cdot nbgen - t} D \quad (4.2)$$

$D$  et  $\eta$  sont des paramètres positifs fixés de façon heuristique. Le problème est de bien choisir ce bruit gaussien. S'il est trop petit, les déplacements dans l'espace sont insuffisants en début de convergence et l'algorithme peut rester bloqué dans un optimum local. Si le bruit est trop fort, l'AG trouvera certainement une zone contenant l'optimum, mais sera incapable de converger localement. Il s'agit donc de diminuer le bruit au fil des générations. Pour ce faire, l'écart type du bruit gaussien  $\eta t$  est mis à jour à chaque itération. Si le bruit est très fort et conduit à ce que l'individu muté ne respecte plus les propriétés des solutions faisables, alors le bruit ne sera pas pris en compte et l'individu ne sera muté.

#### 4.7.5 Maintenir la diversité dans la population

Maintenir une diversité suffisante dans la population permet d'éviter de rester piégé dans les optimaux locaux. Dans cet esprit, les nouveaux individus ne sont pas autorisés à intégrer la nouvelle population si ils y ont déjà des individus clones. L'inconvénient, bien sûr, est la nécessité de comparer chaque individu de la population courante avec le nouveau candidat, ce qui augmente la complexité calculatoire. Mais, dans notre cas d'étude, ce processus comparatif coûte moins de temps que le temps de calcul nécessaire pour évaluer une solution.

Il est également important d'ajuster la puissance de perturbation de la population. Ce processus intervient lorsque le nombre d'échecs dépasse un certain seuil. Le nombre d'échec est le nombre de générations successives dont la solution optimale est restée la même. La puissance de perturbation est calculée comme suit :

$$n_i = \max(0, P) \quad (4.3)$$

tel que,

$$P = \frac{N_t - N_t * \frac{S_t}{S_0}}{\kappa_t} \quad (4.4)$$

- $N_t$  : Nombre total d'individu dans la population courante  $P_t$
- $S_t$  : écart type de la population  $P_t$
- $S_0$  : le plus grand écart type rencontré au cours des générations
- $\kappa_t$  : paramètre proportionnel positif. Ce paramètre augmente au fur et à mesure des générations. Il est proportionnel au nombre de générations.

## 4.8 Algorithme génétique pour l'optimisation de la maintenance préventive non périodique AG-MP(NP)

Il s'agit ici d'un AG pour l'optimisation de la maintenance préventive non périodique. Le système se compose de plusieurs composants. Certaines des caractéristiques des composants, notamment après la première étape d'étude du comportement du système, sont supposées connues.

---

**Algorithme 3** : Paramètres de l' Algorithme Génétique pour l'optimisation de la MPNP

---

```
% Pt : structure de données
% Pt.sol : individus de la population
% Pt.fit : évaluation des individus
% Pt.size : taille de la population
% Pt.nbgen : nombre maximal de générations
% Prcros : probabilité de croisement
% Prmut : probabilité de mutation
% nbechec : nombre de générations successives sans amélioration de
% l'optimum
% diam : diamètre de recherche locale
% indic = {0,1}
% nbmaintmax : vecteur contenant le nombre maximal d'actions de
% maintenance de chaque composant
% mttf : vecteur contenant le temps moyen avant défaillance de chaque
% composant
% bornes : limites des variables
% T : horizon temporel de l'étude de maintenance
```

---

### 4.8.1 Déroulement de l'algorithme

La sélection est responsable du choix des couples qui subiront les opérateurs génétiques de croisement et mutation. L'important est, par rapport aux solutions

choisies, d'échantillonner de façon satisfaisante la population courante, en permettant à ces opérateurs de bien explorer l'espace de recherche. Comme pour l'AG-MP(P) la méthode de sélection employée est la méthode de sélection par tournoi.

**Données :** Modèle d'évaluation de la politique de maintenance

**Résultat :** un calendrier quasi-optimal de MPNP



---

**Algorithme 3** : Algorithme Génétique pour l'optimisation de la MPNP

---

```

t = 0 ;
% Initialisation des paramètres de l'AG
InitialisationAG();
% Génération de la population initiale
Pt.sol = GenerIndiv (Pt.size, indic, diam, nbMaintmax);
% Evaluation et tri de la population
Pt = EvalTri(Pt);
tant que Condition d'arrêt n'est pas satisfaite faire
    % Sélection des individus participant à la reproduction
    Ptsel = Selection (Pt);
    % Croisement
    Ptcro = Croisement (Ptsel, Prcro, nbMaintmax, Pt.size, T, bornes) ;
    % Mise à jour de la probabilité de Mutation
    Prmut = RecuitMutation (Prmut, Ptcro, mttf) ;
    Ptmut = Mutation (Ptcro, Prmut) ;
    % Formation de la nouvelle population
    Pt+1 = CombinaisonTri (Pt, Ptmut) ;
    nbechec = MiseAJour (nbechec, Pt+1) ;
    % Indicateur relatif à l'écart type de la population
    IndEcTy = StdMonotonie (Pt);
    si IndEcTy > ρ alors
        % Améliorer la diversité de la population
        DiversitPop (Pt+1, diam, nbechec)
    fin
    t = t + 1 ;
fin

```

---

### 4.8.2 Croisement

Dès que la sélection est faite, les opérateurs qui génèrent de nouveaux individus sont appliqués : le croisement et la mutation. Étant donné que les solutions sont en codage binaire, l'approche de croisement multi-points s'impose comme la plus évidente. La figure 4.5 illustre son principe. la figure 4.5 illustre un croisement bi-

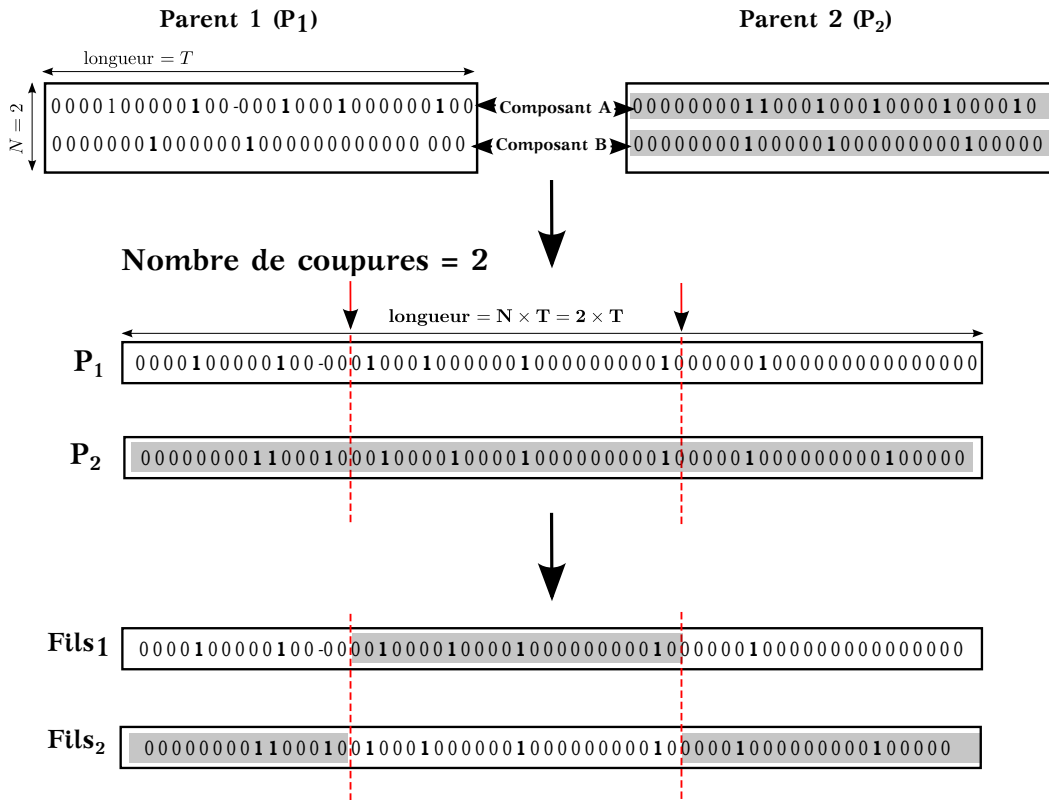


FIGURE 4.5 – Procédure de croisement multi-points

point (nombre de coupures = 2) de deux parents. Chaque Parent est un individu représentant le planning de maintenance préventive non périodique de  $N = 2$  composants. La procédure de croisement commence par concaténer les  $N$  lignes de la matrice du génotype, pour chaque parent, afin de former un vecteur. La difficulté de cette approche réside dans le choix du nombre de points de coupures ainsi que leur position. Parmi les études faites sur ce sujet, nous citons les travaux de [Spears 1991] qui mettent en lumière le rapport entre le nombre de points de coupures et le taux de perturbation de la population et puis le travail de [Jong 1992] qui étend l'étude précédente pour inclure le croisement avec  $n$

points de coupures, le croisement uniforme et la corrélation entre ses différents types de croisement et la taille de la population.

Au début de l'élaboration de cet algorithme, nous avons utilisé un croisement avec 2 points de coupures comme l'illustre la figure 4.5. Pour arriver à ce choix, nous avons comparé la diversité des solutions générées par le croisement en variant à chaque fois le nombre de points de coupures.

Nous avons remarqué alors qu'au delà de 3 points de coupures, la perturbation (l'écart type) de la population reste trop élevée même après plusieurs générations, mais sans pour autant améliorer le temps de convergence. En même temps, opter pour un croisement à 1 point de coupure permet une exploration moins bonne de l'espace de recherche et une convergence prématurée.

Bien que ce type de croisement fonctionne, nous constatons une convergence souvent prématurée de l'algorithme. Nous avons alors choisi d'introduire une modification originale dans la procédure de croisement à savoir le croisement du phénotype illustré dans la figure 4.1. Pourquoi nous ne pas croiser le phénotype au lieu du génotype. Les différentes étapes de cette approche de croisement sont expliquées à partir de l'exemple suivant.

**Exemple 4.8.1** *Un système à deux composants A et B est considéré dans cet exemple. Nous supposons savoir certaines caractéristiques des composants, notamment la fiabilité de chacun. En partant de cette mesure nous pouvons déduire un certain nombre d'informations comme le temps moyen de fonctionnement avant panne (MTTF) pour chaque composant.*

*Dans le cas d'un processus de dégradation exponentiel, le MTTF s'exprime en fonction du taux de défaillance  $\lambda$  comme suit :*

$$MTTF = \frac{1}{\lambda} \quad (4.5)$$

*Il peut aussi être défini en termes de densité de probabilité  $f(t)$  :*

$$MTTF = \int_0^{\infty} tf(t)dt \quad (4.6)$$

avec  $\int_0^{\infty} f(t)dt = 1$ .

*Dans ce travail nous nous plaçons dans un un contexte de maintenance parfaite, ça veut dire que le  $j^{\text{ème}}$  composant du système sera considéré comme neuf après avoir subi une opération de maintenance sous l'hypothèse de l'absence de défauts de jeunesse. Par conséquent, la probabilité qu'un composant tombe en panne entre un moment  $t_1$ , qui succède la date de la maintenance, et le moment  $t_1 + MTTF_j$ , est presque nulle. De ce fait on peut estimer le nombre maximal d'actions de maintenance que peut subir le  $j^{\text{ème}}$  composant sur un horizon temporel  $T$ . Le*

nombre maximal d'actions de maintenance  $\rho_j = \text{nbmaint}_{\max}(j)$  est calculé en fonction du  $\text{MTTF}_j$  de la manière suivante :

$$\rho_j = \lfloor \frac{T}{\text{MTTF}_j - 1} \rfloor = \lfloor \frac{T\lambda_j}{1 - \lambda_j} \rfloor \quad (4.7)$$

Une fois qu'on a une estimation du  $\text{nbmaint}_{\max}^A$  et  $\text{nbmaint}_{\max}^B$ , le travail du croisement commence et s'effectue en trois étapes :

1. Convertir la structure des individus participant au croisement : de génotype en phénotype.
2. Appliquer un croisement BLX- $\alpha$  volumique sur les phénotype
3. Faire la conversion inverse : de phénotype en génotype

Soit  $P_1$  et  $P_2$  deux individus participant au croisement. Le génotype de  $P_1$  comme celui de  $P_2$  est représenté par une matrice binaire. Chaque ligne de la matrice désigne le planning de maintenance d'un composant. Pour  $T = 30$ ,  $\text{MTTF}_A = 7$  et  $\text{MTTF}_B = 10$ , le nombre maximal d'actions de maintenance pour les composant A et B est respectivement 5 et 3. La figure 4.6 explique les différentes étapes de la méthode de croisement.

Il faut préciser qu'avant d'opérer le croisement sur les structures phénotypiques, un réajustement des tailles des phénotypes est essentiel. Compte tenu du nombre maximum d'actions de maintenance des composants  $\text{nbmaint}_{\max}^A$  et  $\text{nbmaint}_{\max}^B$  et dans le cas où pour un composant donné tel A, la longueur du phénotype dépasse  $\text{nbmaint}_{\max}^A$ , alors le phénotype sera tronqué (seuls les premiers  $\text{nbmaint}_{\max}^A$  éléments persistent).

Nous observons cette transformation chez l'individu  $P_2$  au niveau du composant A : la date de maintenance 29 a été supprimé après ce réajustement car le nombre d'actions de maintenance initialement prévu pour le composant A du parent  $P_2$  est supérieur à  $\text{nbmaint}_{\max}^A$ .

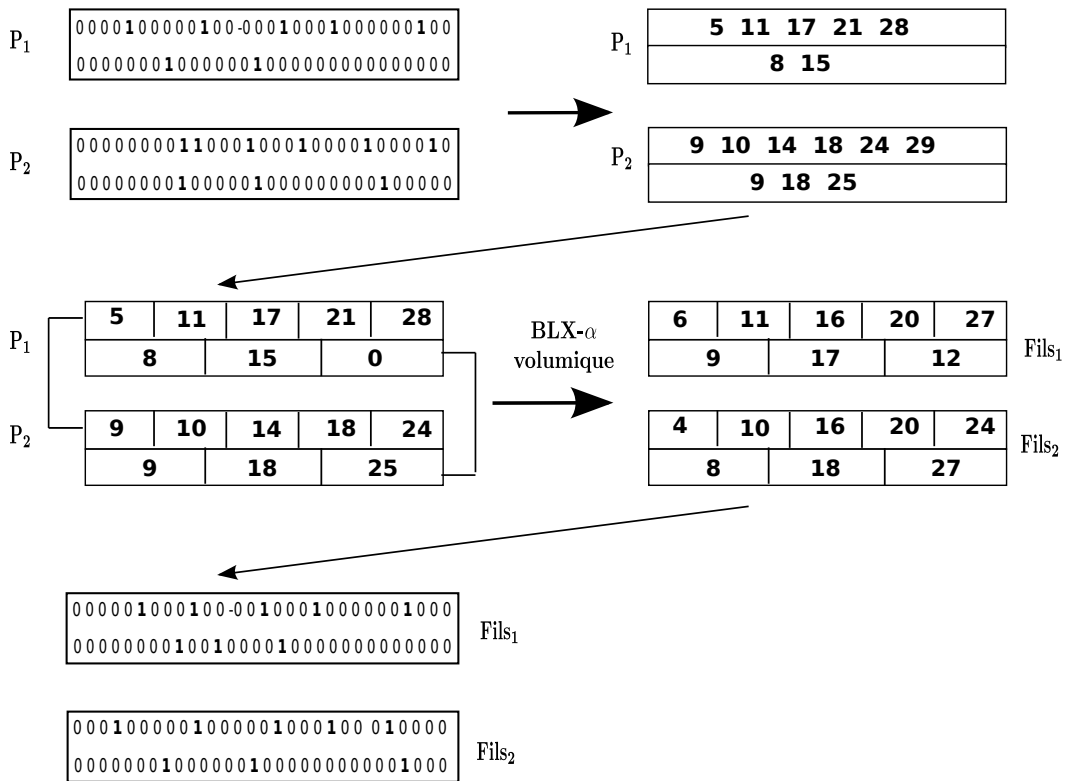


FIGURE 4.6 – Étapes de la procédure de croisement développée pour le AG-MP(NP)

### 4.8.3 Mutation

Comme l'opérateur de croisement, la mutation vise à modifier de manière aléatoire une partie de la population. Elle prend une place de plus en plus importante dans les algorithmes génétiques, parce qu'elle permet essentiellement d'introduire et de maintenir la diversité dans la population de génération en génération et influencer par conséquent les propriétés de convergence. En effet, elle permet d'éviter la dérive génétique : lorsque certains gènes favorisés par un phénomène aléatoire, comme le hasard, se propagent parmi les individus et seront ainsi présents au même endroit sur tous les génotypes. Par ailleurs, elle permet également d'apporter la propriété d'ergodicité : l'AE sera apte à atteindre chaque solution de l'espace de recherche, sans pour autant les parcourir tous durant le processus de résolution [Durand 1994]. De cette façon, l'AE peut converger sans croisement et certaines implémentations fonctionnent de cette manière [Fogel 1966].

Le processus de mutation consiste en l'insertion d'un nouveau matériel génétique dans la population. Généralement, l'opérateur de mutation tire aléatoirement un gène dans le génotype puis le modifie. La mutation est appliquée avec un certain

taux de mutation  $P_{mut}$ , sur l'ensemble de la population. Cette probabilité doit être calibrée convenablement durant toutes les générations pour que la recherche de l'optimum ne soit pas purement aléatoire ou erratique. C'est alors qu'on a entrepris l'étude précise du processus de mutation. En variant le taux de mutation quantitativement et qualitativement, nous proposons trois procédures de mutations. Le mécanisme des deux premières est inspiré de la littérature tandis que la troisième approche, la meilleure, est innovante dans la mesure où elle profite des attributs de notre problème de maintenance pour guider la mutation.

#### 4.8.3.1 Mutation par taux de mutation unique adaptatif

La mutation est utilisée pour ajouter de la puissance exploratoire, mais avec la crainte que l'algorithme ne converge pas. Le taux de mutation est ainsi changé lentement au cours de l'algorithme. Plus fort pendant les premiers stades de l'algorithme de façon à avoir une meilleure exploration, puis il est ensuite atténué afin d'assurer la convergence vers l'optimum.

Cette idée a de fortes similitudes avec l'algorithme de recuit simulé dans lequel il y a un paramètre température qui diminue avec le temps [Lundy 1986]. Sauf qu'ici, le taux de mutation peut connaître parfois des hausses d'une génération à une autre surtout dans les premières générations de l'algorithme. Mais, la tendance globale des valeurs de taux de mutations au fil des générations est décroissante. La procédure de mutation proposée ici et illustrée dans la figure

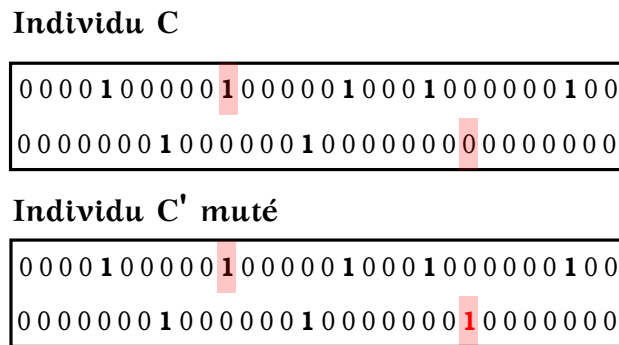


FIGURE 4.7 – Procédure de Mutation adaptative

4.7, consiste à sélectionner aléatoirement un individu puis le modifier avec un taux de mutation adaptatif à tendance décroissante calculé comme suit :

$$P_{mut} = P_{m_0} * \frac{s_t}{s_0} \quad (4.8)$$

$P_{m_0}$  est le taux de mutation initial, fixé au début de l'algorithme,  $s_t$  est l'écart type de la population courante et  $s_0$  est l'écart type de la population initiale.

Les courbes suivantes illustrent le comportement d'un AG-MP(NP) avec une

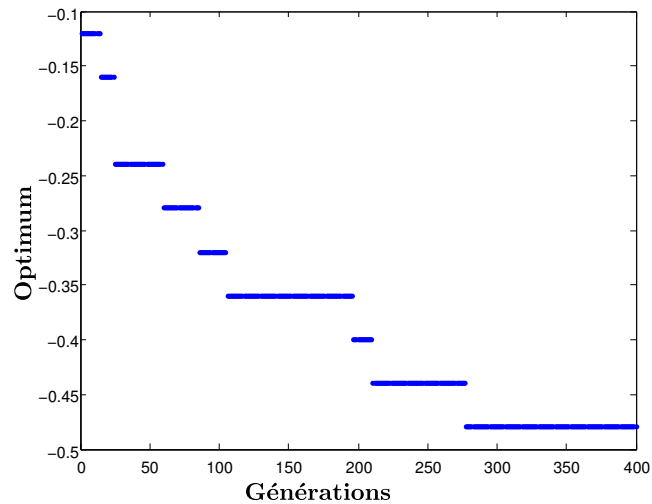


FIGURE 4.8 – Convergence de l'algorithme avec un taux de mutation adaptatif

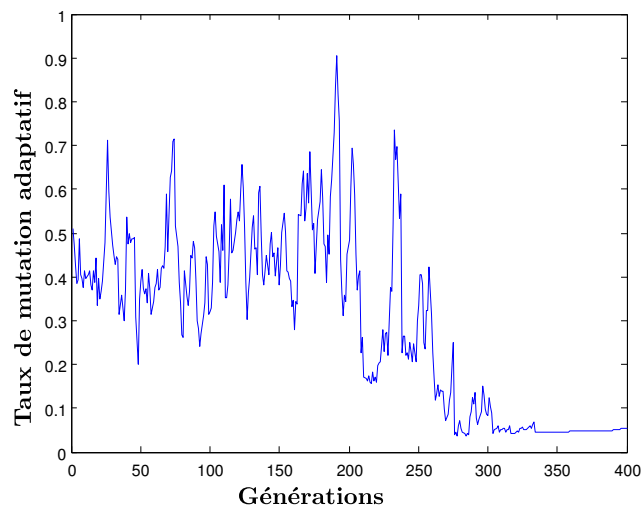


FIGURE 4.9 – Évolution du taux de mutation adaptatif

technique de mutation par taux unique adaptatif. La figure 4.8 montre la convergence de la solution optimale à travers les générations tandis que la figure 4.9 souligne l'évolution du taux de mutation adaptatif. On note que pendant les générations où la solution optimale semble être dans un minimum local (la solution optimale ne varie pas pendant plusieurs générations), le taux de mutation a tendance à augmenter.

#### 4.8.3.2 Mutation adaptative par bit

[Reeves 2002] propose de remplacer un taux de mutation unique, par un taux de mutation distinct pour chaque bit du chromosome. Autrement dit, nous avons une matrice de mutation  $Mut = m_{i,j}$  de même taille que l'individu.  $m_{i,j}$  est la probabilité de mutation du bit qui est à la  $i^{\text{ème}}$  ligne et  $j^{\text{ème}}$  colonne. Cette matrice  $Mut$  est ajustée à chaque pas de temps par le calcul du taux de diversité à chaque position de bit par rapport à toute la population d'individus. On dit que la position de bit  $k$  donnée par le couple  $\{i, j\}$  est suffisamment diversifiée si la proportion de '1' dans cette position est comprise dans un intervalle  $[\delta_1, \delta_2]$ . Si c'est le cas, la baisse de ce taux de mutation élémentaire est renvoyée comme suit.

$$m'_k = \frac{m_k - \xi_{\min}}{2} \quad (4.9)$$

Cependant, dans l'autre cas, le taux de mutation est augmenté légèrement de la manière suivante :

$$m'_k = \frac{m_k + \xi_{\max}}{2} \quad (4.10)$$

tel que  $\xi_{\min} < \xi_{\max}$ . Les bornes  $\delta_1, \delta_2$  sont fixées de façon heuristique et dépendent du contexte de l'application de l'optimisation. Ce type de mutation adaptative par bit se révèle plus efficace que celui présenté juste avant. En fait, dans la plupart des cas d'applications, il donne de meilleurs résultats et diminue le temps de convergence.

#### 4.8.3.3 Mutation adaptative par segment

Il s'agit ici de découper le génotype en segments auxquels nous affectons des taux de mutation différents. Dans notre étude, la délimitation des segments est effectuée sur la base des valeurs de la MTTF et de l'horizon temporel  $T$  de l'étude (tout comme  $nbmaint_{\max}$  d'ailleurs).

Pour chaque gène  $i$  de l'individu, le nombre de segments  $nbsg(i)$  est estimé comme suit :

$$T = a \times MTTF_j + b \quad (4.11)$$

$$nbsg(i) = \begin{cases} a & \text{si } b = 0 \\ a + 1 & \text{sinon} \end{cases} \quad (4.12)$$

où  $a$  et  $b$  sont respectivement le quotient et le reste de la division entière de  $T$  par  $MTTF_j$ . La délimitation des segments pour chaque gène  $i$  se fait de telle sorte que le dernier segment soit de longueur  $MTTF_i$  si  $b = 0$  sinon de longueur  $b$ .

Si on reprend le système présenté précédemment dans l'exemple 4.8.1, la configuration des segments sera comme l'explique la figure suivante : La structure de



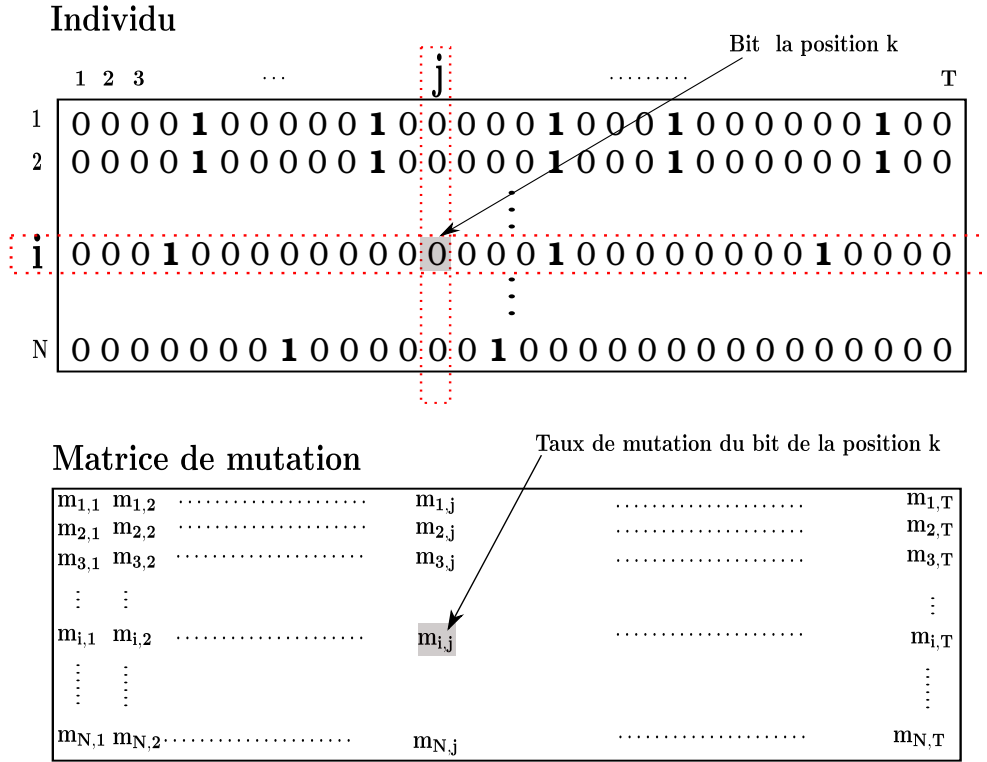


FIGURE 4.10 – Approche de mutation par bit

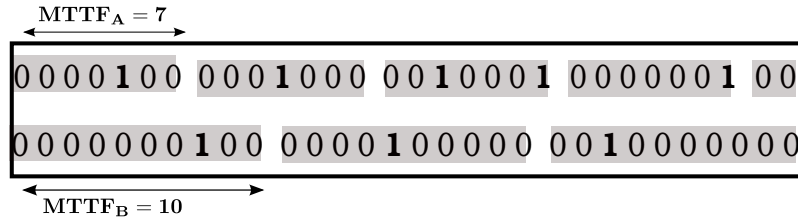


FIGURE 4.11 – Exemple de segmentation d'un individu. A et B deux gènes se rapportant respectivement aux composants A et B du système présenté dans 4.8.1

mutation correspondant à l'individu de la figure 4.11 est :

$$Pr_{mut} = \quad (4.13)$$

$$A : m_{A,1}, m_{A,2}, m_{A,3}, m_{A,4}, m_{A,5}$$

$$B : m_{B,1}, m_{B,2}, m_{B,3} \quad (4.14)$$

Au début de l'algorithme (à  $t = 0$ ) et pour un gène donné  $i$ , les taux de mutation  $m_{i,j}$  ( $j \in 1, \text{nbseg}(i)$ ) des segments  $\text{seg}_{i,j}$  sont égales. Néanmoins, au cours des générations, ces probabilités sont mises à jours en s'adaptant au caractère des morceaux de gènes auxquels elles sont affectées.

Pour juger dans quel sens il faut faire évoluer le taux de mutation  $m_{i,j}$ , on procède à une analyse de diversité du segment  $seg_{i,j}$  dans toute la population assujetti à la mutation.

Cette analyse vise à quantifier la proportion  $Prp$  d'actions de maintenance programmées par ce segment. Suivant la valeur de  $Prp$ , par rapport à deux seuils  $\gamma_1$  et  $\gamma_2$ , fixés, nous faisons fluctuer  $m_{i,j}$ . Le pseudo code 4 suivant détaille la fonction `RecuitMutation()` de l'algorithme 4 dans le cas d'une mutation par segment. Cette fonction inclut l'analyse de diversité et la mise à jour des taux de mutations des segments.

---

**Algorithme 4 : RecuitMutation()**


---

**Données :**

- $P_t^{cross}, Pr_{mut}, mttf$
- $\gamma_1, \gamma_2$
- $\zeta_{min}, \zeta_{max}$ ,

**Résultat :**  $Pr_{mut}$

```

l = Ptcross.size ;
pour chaque gène i faire
    pour chaque segment j faire
        % Extraction des segments
        Sg = ExtractSeg(Ptcross, segi,j) ;
        % Mesure de diversité
        Prp =  $\frac{\sum \sum Sg}{l \times 2}$  ;
        % Adaptation des taux de mutation
        si Prp ∈ [γ1, γ2] alors
            Prmut.mi,j =  $\frac{Pr_{mut}.m_{i,j} + \zeta_{min}}{2}$  ;
        sinon si Prp < γ1 alors
            Prmut.mi,j =  $\frac{Pr_{mut}.m_{i,j} + \zeta_{max}}{2}$  ;
        sinon
            Prmut.mi,j = 1/mttf(i) ;
        fin
    fin
fin

```

---

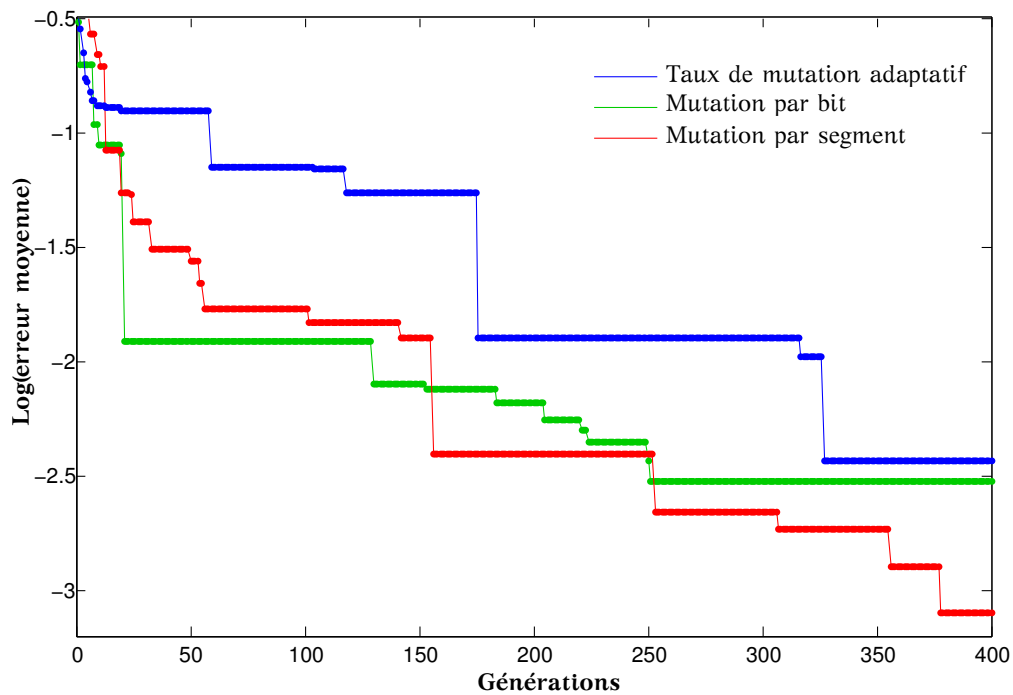


FIGURE 4.12 – Comparaison des graphes de convergence des différentes approches de mutation

#### 4.8.3.4 Comparaison des approches de mutations

Afin de les comparer, nous avons testé les approches de mutation, citées ci-dessous, sur un cas jouet d'optimisation de la maintenance. Il s'agit d'un système à deux composants. Chaque composant se dégrade selon une loi de dégradation donnée. Le système est totalement remplacé après 100 unité de temps. L'objectif est de trouver le calendrier de maintenance des composants qui garantie une disponibilité maximale du système tout en minimisant le coût des interventions.

La figure 4.12 montre les graphes de convergences de l'AG-MP(NP) avec les trois approches de mutations. Chaque graphe de la figure 4.12 représente l'erreur moyenne sur 10 exécutions de l'algorithme, en fonction du nombre d'itérations de l'algorithme.

Le graphe de convergence de l'approche de mutation par segment représente l'erreur moyenne qui décroît le plus rapidement et en quasi-continuité. Toutefois, l'approche de mutation par bit semble être voisine en termes de performances. Nous avons aussi fait évoluer les seuils  $[\delta_1, \delta_2]$  (mutation par bit) et  $[\gamma_1, \gamma_2]$  (mutation par segment) pour voir la sensibilité du temps de convergence par rapport à ces variables.

#### 4.8.4 Maintenir la diversité de la population

Comme dans l'AG-MP(P), le principe de la gestion des doublons est le même. La gestion des clones est assurée principalement lors des étapes de croisement et de mutation.

Par ailleurs, si l'optimum n'est pas encore atteint et la texture de la population reste stationnaire durant un certain nombre de générations successives, on procède à la stimulation de la population par l'introduction de nouveaux individus. Le but de cette démarche est de redonner un "coup de jeune" à la population en l'extirpant de l'optimum local dans lequel elle s'est probablement retrouvée. Mais, ceci ne doit pas négliger l'hypothèse qu'on est peut être proche de l'optimum global. En conséquence, les individus nouveaux doivent être injectés de deux manières différentes.

- Une première famille d'individus doit représenter des solutions candidates générées aléatoirement, explorant tout l'espace de recherche.
- Le deuxième type de famille concrétise un processus de recherche locale. Les individus sont issus de la sphère avoisinant la solution courante. Le diamètre de cette sphère fluctue en concordance avec la progression de l'optimum d'une génération à une autre. Si malgré l'injection de nouveaux individus l'optimum ne s'améliore pas pendant plusieurs générations, la surface de la sphère est alors étirée.

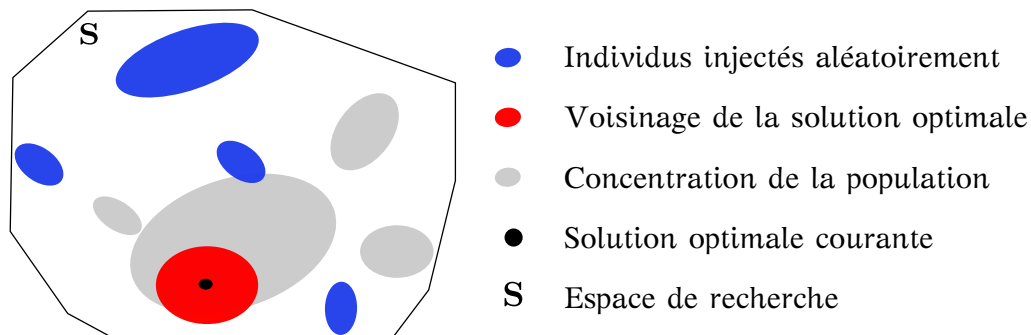


FIGURE 4.13 – Espace de solutions correspondant au deuxième type de famille

#### 4.8.5 Condition d'arrêt

Comme pour toutes les méthodes stochastiques d'optimisation, on ne peut pas identifier de condition d'arrêt simple des algorithmes évolutionnaires car il n'y a pas de solution rigoureuse qui permette de dire si les optimums trouvés sont

corrects. Néanmoins, aujourd'hui, la littérature propose plusieurs critères d'arrêt. Son choix est important car un critère peu performant peut conduire à de nombreuses évaluations inutiles de la fonction objectif. Les critères les plus connus sont :

1. Le temps imparti est écoulé. Les algorithmes se terminent après un nombre de générations fixé a priori.
2. Les algorithmes sont arrêtés si la valeur de la fonction objectif ne s'améliore plus au bout d'un certain nombre de générations. Toutefois, ce critère n'est pas toujours pertinent car on observe souvent que la valeur de la fonction objectif ne s'améliore qu'après un grand nombre de générations. Ce problème apparaît souvent dans le cas où la fonction objectif a une forme de type escalier.
3. Le meilleur individu de la dernière génération atteint une valeur fixée a priori (meilleur). Dans certains cas, la valeur optimale peut être estimée ou connu (s'il s'agit par exemple d'un benchmark). Dans ce cas, l'algorithme s'arrête dès que la valeur de la fonction objectif du meilleur individu est aussi bonne que la valeur cible. Ce critère est principalement utilisé pour tester les performances des algorithmes avec des problèmes d'optimisation dont les solutions sont connues.
4. Performances en ligne et hors ligne [Hajji 2003]. Durant l'exécution de l'algorithme génétique, il est indispensable de disposer d'outils et de statistiques pour suivre la dynamique d'évolution de la population à travers les générations. Ces mesures statistiques sont utilisées dans [Grefenstette 1986] et [Hajji 2003]. Nous citons la performance en ligne et la performance hors ligne.

**Performance en ligne :** La performance en ligne est estimée par la mesure de la performance moyenne de l'algorithme, basée sur la fonction objectif :

$$\text{Performance en ligne(NE)} = \frac{1}{\text{NE}} \sum_{n=1}^{n=\text{NE}} F(n) \quad (4.15)$$

NE est le nombre total d'évaluations de la fonction objectif jusqu'à l'évaluation courante,  $F(n)$  est la  $n^{\text{ième}}$  évaluation de la fonction objectif.

Lors de l'exécution de l'AG, la Performance en ligne converge vers une valeur stable (stationnaire).

**Performance hors ligne :** La mesure de la performance hors-ligne, ressemble à celle de la performance en-ligne. Elle accorde plus d'importance

aux performances des meilleurs individus.

$$\text{Performance hors ligne(NE)} = \frac{1}{\text{NE}} \sum_{n=1}^{n=\text{NE}} F_{\max}(n) \quad (4.16)$$

avec  $F_{\max}(n) = \sup_{\beta=1\dots n} F(\beta)$ .  $F_{\max}(n)$  enregistre la meilleure fonction d'adaptation jusqu'à l'instant présent et ignore toute autre évaluation. Par conséquent, si la performance hors ligne converge vers une valeur stable durant l'évolution, la chance d'obtenir une solution encore meilleure diminue fortement, et continuer l'algorithme serait une perte de temps.

5. convergence des bits (CB). On peut juger la convergence et l'observer directement sur les bits des individus. Ce critère est valable seulement dans le cas d'un codage binaire. On dit qu'un bit converge si 95 % de la population partage la même valeur [Hajji 2003]. La population converge lorsque tous les bits ont convergé. Le calcul de la convergence d'un code binaire est basé sur les valeurs des bits à chaque position.

Il est évident que le critère d'arrêt numéro 1 (nombre maximal de générations), seul, est insuffisant pour détecter la convergence de l'AG. Deux cas de figures sont discernée :

- soit l'optimisation continue inutilement entraînant une perte de temps, alors que l'optimum a été trouvé;
- soit l'arrêt est prématuré.

Le critère de convergence 2, basé sur la variation de la valeur de performance du meilleur individu à travers les générations, est également insuffisant à cause du caractère stochastique de l'AG. Le critère de convergence numéro 5 ne serait pas à son tour efficace seul, car la convergence de toute la population vers un seul point n'est pas toujours achevée. La convergence avec les critères d'arrêt de type 4 ne peut pas être détectée qu'avec un retard variable rattaché au moment pour lequel la valeur de performance devient stable.

En partant de ce constat, nous proposons d'utiliser, dans nos travaux, certains de ses critères combinés ensemble ou seuls pour détecter le plus tôt possible la convergence de l'algorithme.

Selon le contexte du problème et la disponibilité de certaines informations on peut privilégier l'usage d'un critère par rapport à un autre. En effet le condition d'arrêt numéro 3 pourrait être envisageable si on disposait d'informations (tels les exigences budgétaires et sécuritaires pour un problème d'optimisation) sur la valeur optimale (ou du moins sur le seuil d'acceptabilité) de la fonction objectif.

En l'absence de toute visibilité sur l'allure et sur le mérite de la fonction d'adaptation, nous opterons pour l'utilisation de certains des autres critères alliés ensemble. Dans cette optique, nous pouvons agencer par exemple les critères 1 et 2

ou 1 et 4.

Cependant, c'est une toute nouvelle combinaison qui nous a donnée le plus de satisfaction. Il s'agit de conjuguer le critère 1 (nombre maximum de génération) avec une mesure de disparité de la population. Cette dernière, nous éclaire sur la texture de la population. De ce fait, si 97 % de la population ne changent pas pendant un certain nombre de générations, malgré les méthodes de diversification de la population, et si la condition du critère 1 est satisfaite, l'algorithme s'arrête.

## 4.9 Recuit simulé modifié (RSM) pour l'optimisation de la maintenance préventive

La lenteur de la convergence de l'algorithme du recuit simulé standard reste un inconvénient dans son application pour notre problème d'optimisation. Dans cette section nous proposons une implémentation modifiée du recuit simulé accélérant sa convergence qu'on va appliquer à notre problème d'optimisation de la maintenance.

### 4.9.1 Déroutement de l'algorithme

L'algorithme du recuit simulé exploite la règle de *Metropolis* (voir algorithme 5) qui permet de décrire le comportement d'un système "en équilibre thermodynamique" à une certaine température  $T$ .

Partant d'une configuration donnée, on fait subir au système (une solution) une modification élémentaire; si cette modification a pour effet de diminuer la fonction objectif (où énergie), elle est acceptée; si elle provoque au contraire une augmentation  $\Delta F$  de la fonction objectif, elle peut être acceptée tout de même, avec la probabilité  $e^{-\frac{\Delta F}{T}}$ .

On itère ensuite ce procédé, en gardant la température constante, jusqu'à ce que l'équilibre thermodynamique soit atteint; concrètement au bout d'un nombre suffisant de modifications ou d'itérations. On diminue alors la température, avant d'effectuer une nouvelle série de transformations : la loi de décroissance par paliers de la température est souvent empirique, tout comme le critère d'arrêt du programme.

Nous présentons dans l'algorithme 6 le pseudo code de l'algorithme du recuit simulé modifié.

**Algorithme 5** : Règle de Metropolis( $x, x', T, rej$ )

---

```

si  $\Delta F = f(x') - f(x) \leq 0$  alors
   $x'$  solution acceptée et sauvegardée ;
   $rej \leftarrow 0$  ;
  sinon
     $x'$  solution acceptée avec la probabilité  $P = \exp(-\frac{\Delta F}{T})$  ;
    On génère un nombre aléatoire  $r \in [0, 1]$ ;
    si  $r \leq P$  alors
       $x'$  acceptée et  $x \leftarrow x'$  ;
       $rej \leftarrow 0$  ;
      sinon
         $x'$  rejetée ;
         $rej \leftarrow rej + 1$ 
      fin
    fin
  fin
fin

```

---

**4.9.2 Paramètres**

La principale difficulté dans la mise en place de l'algorithme du recuit simulé est le réglage des paramètres initiaux, du schéma de refroidissement, des procédures de recherche de solutions et du critère d'arrêt de l'algorithme.

**Paramètres de contrôle relatifs à la température**

Cet ensemble inclut essentiellement la température initiale  $T_0$ , la température minimale  $T_{\min}$  et la longueur du palier (le nombre d'itérations à température constante).

La température initiale peut être fixée arbitrairement ou bien à la suite d'un prétraitement qui tient compte de plusieurs facteurs tels que la probabilité d'acceptation de la plus mauvaise solution et l'accroissement maximal autorisé de la fonction objectif ([Kirkpatrick 1983]). Elle doit être suffisamment élevée pour que la descente en température soit aussi lente que possible.

La longueur du palier (ou équilibre thermodynamique) doit être assez élevée pour atteindre le quasi-équilibre thermodynamique et, en même temps, suffisamment faible pour réduire le temps d'optimisation. Sa valeur est régie par les deux paramètres suivants :

- Le paramètre qui borne le nombre d'itérations maximal par palier



---

**Algorithme 6 :** Algorithme du recuit simulé modifié

---

```

1. Initialisation( $x_0, T_0, \text{rej}, \mathcal{P}^{\text{desc}}$ ) ;
2.  $T \leftarrow T_0$ ;  $x \leftarrow x'$ ;  $\text{bst} \leftarrow x$  ;
3.  $x' = \text{NouvelleSol}(x, \text{rej}, \text{bst})$  ;
4.
si Non ( Verif ( $x', \mathcal{P}^{\text{desc}}$ ) ) alors
    |   Retourner à l'étape 3 ;
fin
5.  $f(x') = \text{eval}(x')$  ;
6.  $\Delta F = f(x') - f(x)$  ;
7. Règle de Metropolis( $x, x', T, \text{rej}$ ) ;
8.  $\text{bst} = \text{Ajour}(\text{bst}, x)$ ;
9.
si Equilibre Thermodynamique atteint alors
    |   % Système figé
    |   si Critère arrêt( $\text{rej}, T, x$ ) alors
    |       |   Fin algorithme ;
    |       |   sinon
    |       |       |   % Diminution lente de T
    |       |       |    $T = \lambda T$ ;
    |       |       |   Retourner à l'étape 3 ;
    |       |       fin
    |       fin
    |   sinon
    |       |   Retourner à l'étape 3 ;
    |       fin
fin

```

---

– Le paramètre qui borne le nombre de changements (mouvements de la solution) à température constante.

**Loi de décroissement de la température**

Elle doit être bien définie dans la mesure où les températures doivent être assez

hautes pour que l'algorithme évite les minimums locaux et en même temps assez basses pour assurer la convergence de l'algorithme et réduire le temps de calcul. Le changement de température est effectué au moment où l'équilibre thermodynamique est atteint, selon la fonction de décroissement linéaire suivante :

$$T = \lambda T \quad (4.17)$$

avec  $0 < \lambda < 1$ ; Dans nos essais,  $\lambda$  varie entre 0.4 et 0.9. Cette loi permet d'avoir un décroissement rapide au début de l'optimisation et suffisamment lent proche de la convergence.

#### **Stratégies de recherche de voisinage**

La structure de recherche de voisinage est la voie pour produire de nouvelles solutions différentes de la solution courante. La qualité des solutions dépend étroitement des procédures de recherche. Pour améliorer la vitesse de convergence du RS, nous proposons deux stratégies de recherche de solutions. Ces dernières se concluent par une procédure de vérification de la faisabilité de la solution. Cette procédure prend en considération les contraintes liées aux différentes variables de la solution.

#### **Stratégie de recherche avec mémoire**

Dans cette stratégie une fonction mémoire est instaurée. Cette fonction mémorise l'ensemble des paramètres de la solution dont la fonction objectif est la plus petite depuis le début de l'algorithme. Cette solution sera utilisée comme paramètre d'entrée dans la procédure de recherche d'une nouvelle solution, si le nombre de rejets successifs de solutions **rej** dépasse un certain seuil  $\varepsilon_{rej}$ .

#### **Stratégie de recherche avec mémoire et structures de recherche de voisinage différentes**

Cette stratégie combine la technique de la fonction mémoire avec deux méthodes de recherche de voisinage. Une première méthode de recherche est appliquée par défaut. Si au bout d'un nombre  $\varepsilon_{rej}$  de rejets successifs (**rej** =  $\varepsilon_{rej}$ ), la solution retenue par la fonction mémoire est sollicitée afin de créer une nouvelle solution au moyen de la deuxième méthode de recherche de voisinage.

#### **Critère d'arrêt de l'algorithme**

Un grand nombre de critères existent. Le processus d'optimisation est arrêté quand aucune amélioration sensible n'est réalisée ( $rej > \varepsilon_{rej\max}$ ), où quand le paramètre de contrôle est inférieur à une certaine valeur  $T_{\min}$ .

### 4.9.3 Résultats

Pour valider les modifications apportées au RS nous avons utilisé le benchmark *six-hump camelback* (cf.figure 4.14) à deux dimensions (quatre minima locaux ainsi que deux minima globaux sont observés pour cette fonction).

$$f(\mathbf{x}) = (4 - 2.1x_1^2 + \frac{x_1^4}{3})x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2 \quad (4.18)$$

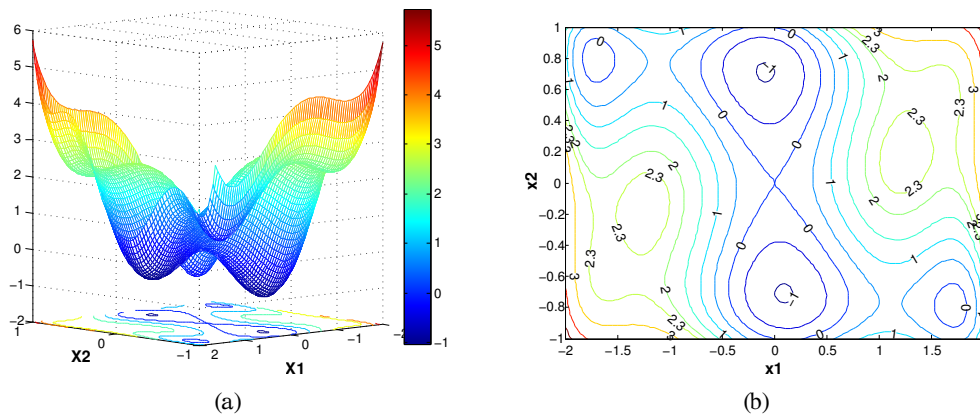


FIGURE 4.14 – Fonction Six-hump Camel Back et le tracé de contours

Afin de choisir la meilleur stratégie de recherche de voisinage nous avons comparé les performances du RSM en changeant à chaque fois de stratégies.

Trois techniques ont alors été implémentées avec le RSM.

- **Technique 1** : n'utilise pas de fonction mémoire et possède une seule méthode de recherche de voisinage.
- **Technique 2** : correspond à la description de la stratégie de recherche avec mémoire.
- **Technique 3** : utilise une fonction mémoire et deux méthodes de recherche de voisinage.

D'après la figure 4.15, nous remarquons que la technique 1 donne les moins bon résultats en termes de temps de convergence contrairement à la technique 3. La technique 2 quant à elle se situe en deuxième position.

On peut déduire alors qu'avoir plusieurs structures de recherche de voisinage combinées à une fonction mémoire améliore significativement le temps de convergence.

Nous illustrons la figure dans 4.16 l'évolution des paramètres de la solution optimale pour chaque technique. Nous observons qu'avec la technique 3 (voir figure 4.16c) la recherche se focalise plus rapidement dans la zone du voisinage proche

de l'optimum contrairement à la technique 1 (voir figure 4.16a) où la recherche paraît plus lente.

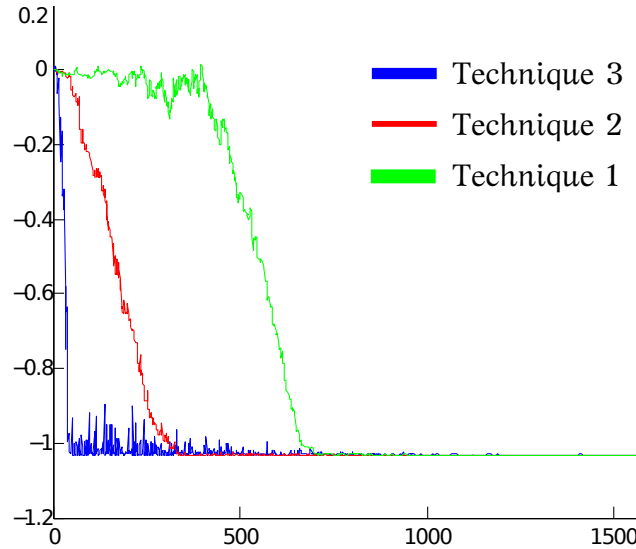


FIGURE 4.15 – Vitesse de convergence des trois techniques de recherche de voisinage

#### 4.10 L'algorithme ev-MMOGA pour l'optimisation de la maintenance préventive

Le ev-MMOGA est un algorithme évolutionnaire élitiste multi-objectif [Coello 2005], basé sur une version de l'algorithme ev-MOGA de [Martínez-Iranzo 2009].

Il s'appuie principalement sur le concept de  $\varepsilon$ -dominance [Laumanns 2002] et s'articule autour de trois populations courantes.

- $P(t)$  : population principale.
- $A(t)$  : population archive où sont stockées les solutions Pareto (résultat de l'optimisation).
- $G(t)$  : population auxiliaire.

Le but de l'algorithme est d'assurer la convergence de la population  $A(t)$  vers un ensemble de solutions  $\varepsilon$ -Pareto,  $\Theta_p^*$  convenablement distribuées le long du front de Pareto  $F(\Theta_p)$ .

Avant de définir le concept d' $\varepsilon$ -dominance, nous devons présenter tout d'abord la majeure particularité principale de l'algorithme qui est la notion de subdivision de l'espace des objectifs.

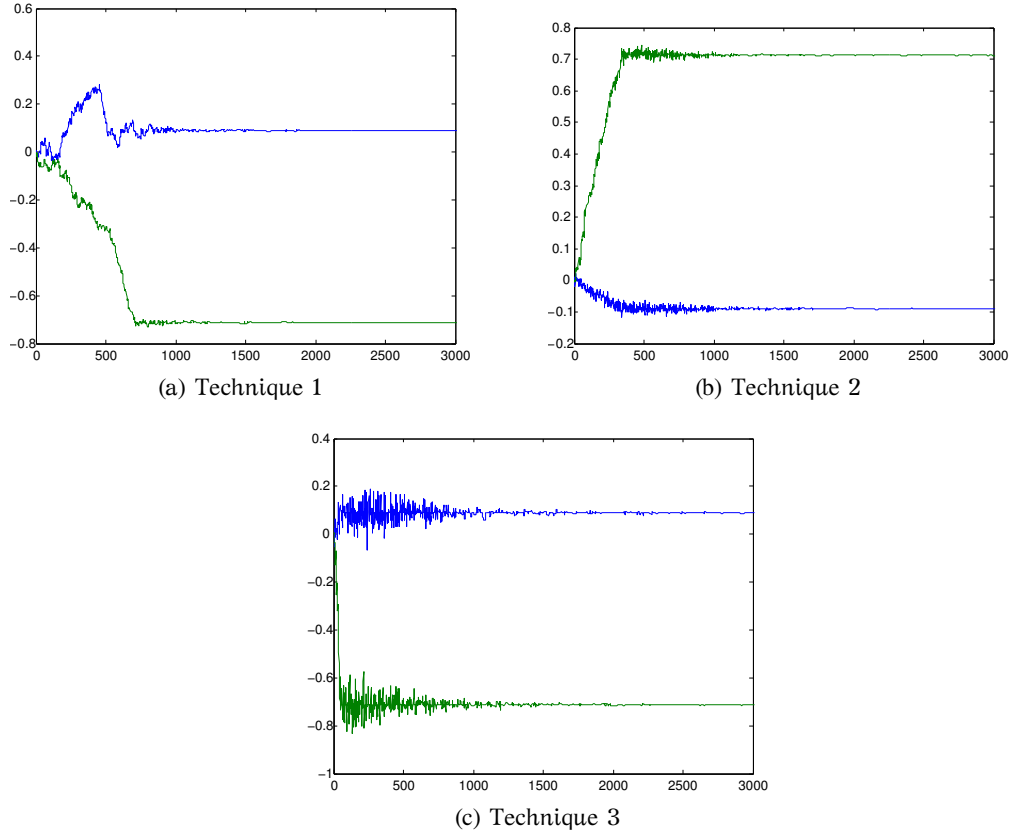


FIGURE 4.16 – Représentation de l'évolution des paramètres solutions

#### 4.10.1 Subdivision dynamique de l'espace des objectifs

La règle ici est de subdiviser l'espace des objectifs, produisant de la sorte, une grille pour chaque dimension. Chaque dimension  $i$  de l'espace des objectifs,  $i \in \mathcal{O} = [1 \dots k]$ , est subdivisée en un nombre  $n_{\text{div}_i}$  de divisions (appelées aussi cases ou mailles) constituant une grille de résolution  $\varepsilon_i$  tel que :

$$\varepsilon_i = \frac{f_i^{\max} - f_i^{\min}}{n_{\text{div}_i}} \quad (4.19)$$

$$f_i^{\max} = \max_{X \in \Theta_{\mathbf{p}_\varepsilon}^*} f_i^{\max}(X), \quad f_i^{\min} = \min_{X \in \Theta_{\mathbf{p}_\varepsilon}^*} f_i^{\min}(X) \quad (4.20)$$

Cette grille préserve la diversité du Front de Pareto (approximé)  $F(\Theta_{\mathbf{p}_\varepsilon}^*)$  puisque chaque case ne peut être occupée que par une seule solution provenant de  $A(t)$ .

Dans la version originale de l'algorithme, les  $n_{\text{div}_i}$  étaient fixes pour tout l'algorithme.

Une des améliorations apportées par le ev-MMOGA est de rendre dynamique la subdivision. De ce fait, à chaque itération de l'algorithme, les  $n_{div_i}$  changent en s'adaptant mieux aux caractéristiques du problème. Nous détaillerons cette adaptation dans la section 4.10.4.

Il est à noter que cette subdivision permet d'ajuster dynamiquement les limites du front de Pareto et d'éviter ainsi la perte des solutions se trouvant aux extrémités du front de Pareto. La figure 4.17 montre un exemple de subdivision d'un problème bi-objectifs.

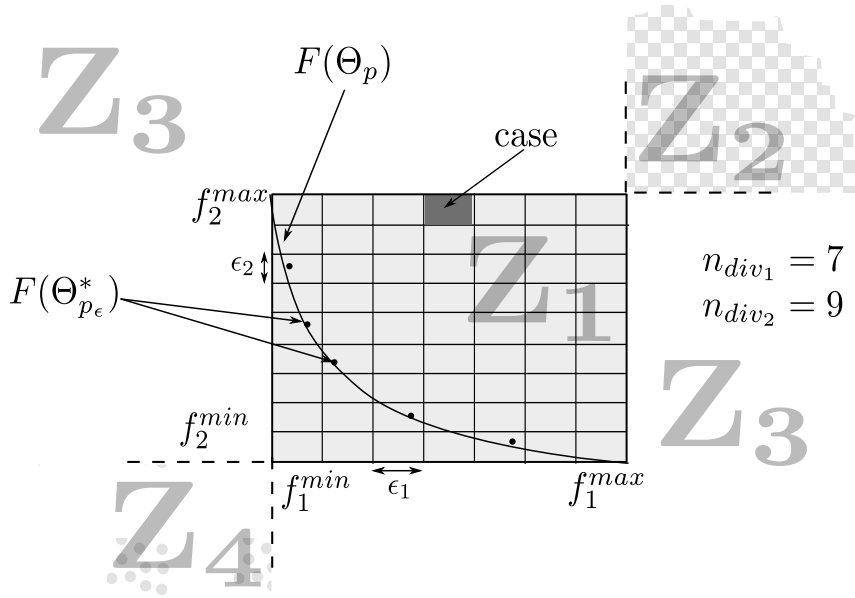


FIGURE 4.17 – Subdivision de l'espace des objectifs. Le Front  $\epsilon$ -Pareto  $F(\Theta_{p_\epsilon}^*)$  d'un problème bi-objectif.  $f_1^{\min}$ ,  $f_2^{\min}$ ,  $f_1^{\max}$  et  $f_2^{\max}$  sont les limites du front de Pareto.  $\epsilon_1$  et  $\epsilon_2$  correspondent aux largeurs des cases (divisions).  $n_{div_1}$ ,  $n_{div_2}$  sont les nombres de divisions de chaque dimension.

#### 4.10.2 Le concept de $\epsilon$ -dominance

Le concept d' $\epsilon$ -dominance est défini dans ce qui suit.

$\forall i \in \mathcal{O}$ , pour une solution  $X \in X^n$ ,  $div_i(X)$  est définie par :

$$div_i(X) = \left\lceil \frac{f_i(X) - f_i^{\min}}{f_i^{\max} - f_i^{\min}} \cdot n_{div_i} \right\rceil \quad (4.21)$$

**Définition 4.10.1**  $\varepsilon$ -dominance

soit  $\text{div}(X) = \{\text{div}_1(X), \dots, \text{div}_k(X)\}$ .

$X^1$  et  $X^2$  deux solutions ayant respectivement les mérites  $F(X^1)$  et  $F(X^2)$ .

On dit que  $X^1$   $\varepsilon$ -domine  $X^2$  :

$$\begin{aligned} X^1 <_{\varepsilon} X^2 &\Leftrightarrow \text{div}(X^1) < \text{div}(X^2) \\ &\Leftrightarrow \text{div}(X^1) = \text{div}(X^2) \text{ et } X^1 < X^2 \end{aligned}$$

Par conséquent,  $\Theta_{p_{\varepsilon}}^* \subseteq \Theta_p$  est  $\varepsilon$ -Pareto  $\Leftrightarrow$

$$\forall X^1, X^2 \in \Theta_{p_{\varepsilon}}^*, X^1 \neq X^2, \text{div}(X^1) \neq \text{div}(X^2) \text{ and } \text{div}(X^1) \not<_{\varepsilon} \text{div}(X^2) \quad (4.22)$$

### 4.10.3 Déroulement de l'algorithme

L'algorithme commence comme l'AG mono-critère, c'est-à-dire que les spécifications initiales du processus sont stipulées (probabilités de croisement et de mutation, taille de la population et nombre maximal de générations).

On doit connaître au préalable : le nombre de variables de chaque individu, les limites acceptables de chaque variable et le nombre d'objectifs qui seront pris en compte.

Une description de l'algorithme ev-MMOGA est donnée dans l'algorithme 7 comme suit : Avec les spécifications initiales, on crée d'abord une population initiale de solutions possibles du problème, appelée  $P(t)$ . Les individus sont générés aléatoirement dans les limites préétablies. Les individus de  $P(t)$  sont par la suite évalués en fonction du problème posé (fonction objectif).

#### Formation de l'archive

Cette procédure se résume en trois étapes :

1. Détection des solutions non dominées.
2. Calcul des limites du front de Pareto
3. Formation du front  $\varepsilon$ -Pareto

La vérification (ou non vérification) de la condition de Pareto 3.3.1 partage la population en deux groupes : un groupe formé par les solutions non-dominées  $\Theta_{ND}$  ; et un autre formé par les solutions dominées  $\Theta_D$ .

A partir des solutions non-dominées  $\Theta_{ND}$ , les limites du front de Pareto  $\{f_i^{\max}, f_i^{\min}, \forall X \in \Theta_{ND}\}$  sont calculées comme le stipule l'équation 4.20.

**Algorithme 7 : ev-MMOGA****Résultat : un calendrier quasi-optimal de MP**


---

```

t = 0 ;
Initialisation-ev-MMOGA(ndivi);
Pt = GenerPop (Xn);
evaluation(P(t));
% Formation de l'archive
A(t) = Archiver(P(t), A(t)) ;
tant que (t < Pt.nbgen) faire
    % Création de la population intermédiaire
    G(t) = Reproduction(P(t), A(t)) ;
    evaluation(G(t)) ;
    % Adapter le nombre de divisions
    Adaptation(ndivi);
    % Mise à jour de l'archive
    A(t + 1) = MettreAJourArchive(G(t), A(t)) ;
    % Mise à jour de la population
    P(t + 1) = MettreAJourPop(P(t)) ;
    t = t + 1 ;
fin

```

---

Cette dernière opération est indispensable pour la faisabilité de l'étape suivante. Dans cette dernière étape,  $\Theta_{ND}$  est évaluée en fonction de l' $\varepsilon$ -dominance : les individus qui ne sont pas  $\varepsilon$ -dominés par les individus de  $A(t)$  sont inclus dans  $A(t)$ . Le but est d'obtenir une répartition régulière des individus sur la frontière Pareto. A l'issue de cette évaluation, la population archive  $A(t)$  est formée. Sa taille, notée,  $A(t).size$  est variable mais ne peut pas excéder le seuil suivant :

$$|\Theta_{p_\varepsilon}^*| \leq \frac{\prod_{i=1}^k n_{div_i} + 1}{n_{div_{max}} + 1}, \quad n_{div_{max}} = \max_i n_{div_i} \quad (4.23)$$

**Création de la population intermédiaire**

Pour assurer la pérennité de la population, la phase de la reproduction est indispensable. Cela commence bien évidemment par la sélection d'un couple de reproducteurs. On prend aléatoirement un individu  $X^A$  de  $A(t)$  et un autre  $X^P$  de



$P(t)$  qu'on essaye de combiner/ muter selon une certaine probabilité fixée préalablement au début de l'algorithme.

Jusqu'ici nous n'avons pas précisé s'il s'agit d'un problème d'optimisation de la MPP ou bien de la MPNP, car les étapes précédentes de l'ev-MMOGA pourraient bien s'appliquer dans un cas que dans un autre, car la spécification des méthodes de croisement et de mutation dépendent étroitement de la forme et du type des solutions. Ces opérateurs donnent naissance à de nouveaux individus (enfants) basés sur les informations contenues dans les individus courants (parents), de manière à explorer efficacement l'espace de recherche. Nous détaillerons ces opérateurs dans les sections 4.10.5 et 4.10.5.3.

#### **Adapter le nombre de divisions**

Dans cette procédure, le nombre de divisions par espace d'objectif est repensé.

Ce changement se répercute bien évidemment en premier lieu, sur les valeurs des  $\varepsilon_i$ , ce qui donne un nouveau visage à l'espace des objectifs (nouvelle répartition). La deuxième conséquence de ce changement porte sur l'évaluation des individus de  $A(t)$  par rapport à l' $\varepsilon$ -dominance.

#### **Mise à jour de l'archive**

Cette fonction va juger si un individu  $X^G$  de  $G(t)$  mérite ou non d'intégrer l'archive  $A(t)$ . Si on prend un problème d'optimisation à deux objectifs, en traçant les limites du front de Pareto  $\{f_1^{\max}, f_1^{\min}\}$  pour le premier objectif et  $\{decef_2^{\max}, f_2^{\min}\}$  pour le deuxième, nous pourrions départager l'espace des objectifs comme le montre la figure 4.17.

- Si  $F(X^G)$  appartient à la zone  $Z_1$  et n'est pas  $\varepsilon$ -dominé par aucun des individus de  $A(t)$ , alors  $X^G$  peut intégrer  $A(t)$ . Si la case est déjà occupée par une solution qui n'est pas  $\varepsilon$ -dominée, alors l'individu se trouvant le plus au centre de la case est gardé. Les individus de  $A(t)$  qui sont  $\varepsilon$ -dominés par  $X^G$  sont éliminés.
- Si  $F(X^G)$  appartient à la zone en damier  $Z_2$ ,  $X^G$  ne sera pas inclu dans l'archive (Il est dominé par tous les individus de  $A(t)$ ).
- Si  $F(X^G)$  appartient à la zone non colorée ( $Z_3$ ), on se retrouve dans le cas où nous devons redéfinir les limites du front de Pareto qui doit certainement changer de forme et par conséquent . On retourne à l'étape "Formation de l'archive" avec les paramètres  $(A(t) \cup X^G, \phi)$ .
- Si  $F(X^G)$  est dans la zone  $Z_4$ , tous les individus de  $A(t)$  seront rejetés puisqu'ils sont tous  $\varepsilon$ -dominés par  $X^G$ . Ce dernier intégrera  $A(t)$  et  $F(X^G)$  représentera les limites de l'espace des objectifs.

#### **Mise à jour de la population**

Le but de cette étape de l'algorithme est de remplacer certains individus de la

population courante  $P(t)$  par d'autres meilleurs, afin de former la population du future  $P(t+1)$ .

Les individus qui remplaceront des mauvais éléments de  $P(t)$  proviennent de la population intermédiaire  $G(t)$  fruit de l'accouplement de la population courante et de la population Archive  $A(t)$ .

Une solution  $X^G$  de  $G(t)$  remplacera un individu  $X^P$ , si ce dernier existe et est sélectionné aléatoirement parmi les individus de  $P(t)$  qui sont dominés par  $X^G$ .

#### 4.10.4 Choix et adaptation d' $\varepsilon$

Une bonne prise en compte de l'espace des objectifs dépend étroitement du choix du nombre de divisions pour chaque objectif.

Initialement, les valeurs des  $n_{div_i}$ ,  $i \in \mathcal{O}$  peuvent être choisies arbitrairement, mais avec prudence, car un mauvais maillage initial peut altérer fortement la qualité de convergence.

En effet, une très petite valeur de  $n_{div_i}$  au début de l'algorithme, pourrait causer la perte de beaucoup de solutions Pareto optimales, car le nombre de divisions n'était pas en concordance avec le nombre souhaité de solutions dans le front de Pareto.

Pour corriger ce point, on peut penser à augmenter cette valeur dans les prochaines générations de l'algorithme, mais ce n'est malheureusement pas la solution idéale. Une augmentation du maillage au cours de l'algorithme représenterait une considérable perte de temps dans la mesure ou le nombre d'évaluations augmenterait.

Cela revient en fin de compte à relancer l'algorithme depuis le début mais avec un  $n_{div_i}$  plus grand. Ceci implique alors une nouvelle recherche des solutions Pareto-optimales, alors que beaucoup d'entre elles ont été perdues lors des premières itérations de l'algorithme (une seule solution est permise à la fois dans une case).

Un nombre extrêmement élevé de mailles risque d'entraîner un ensemble Pareto très vaste (surtout pour des problèmes à variables continues).

La permanence d'un nombre excessif de ces solutions dans le processus évolutif aboutit à la diminution de la diversité, et peut entraîner une convergence prématurée.

Pour réussir cette étape d'initialisation des  $n_{div_i}$ , l'idéal, serait de tenir compte des remarques évoquées ci-dessus et de ne pas exclure de prendre l'avis d'un expert qui pourrait cerner encore mieux l'espace des objectifs. Pour un problème

d'optimisation de la maintenance, il s'agit souvent de minimiser une fonction coût et maximiser une fonction gain. Un expert du domaine pourrait alors nous donner une échelle pour chaque grandeur, ce qui donne une indication sur les valeurs minimales à prendre pour les  $n_{\text{div}_i}$ .

Dans notre travail, nous considérons généralement, des valeurs initiales assez importantes de  $n_{\text{div}_i}$ , qu'on tend par la suite à ajuster au fur et à mesure de l'évolution de la population de l'archive  $A(t)$ .

Cet ajustement contribue à l'amélioration de l'établissement de la frontière Pareto, en évitant une convergence prématurée ou encore des agglomérations de solutions dans certaines régions comme le montre la figure 4.18.

Pour chaque espace d'objectif, la valeur de  $\varepsilon_i$  est proportionnelle à  $n_{\text{div}_i}$  ce qui

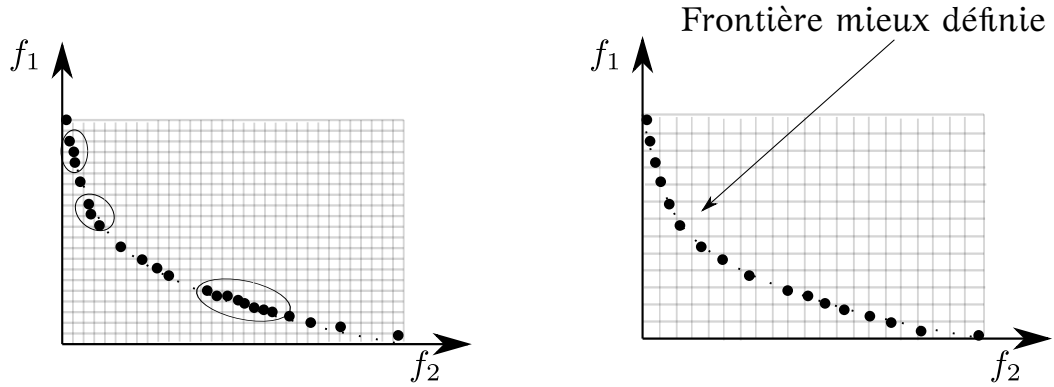


FIGURE 4.18 – Décongestion réalisée dans l'espace des objectifs

fait que si  $n_{\text{div}_i}$  diminue,  $\varepsilon_i$  augmente.

Une adaptation dynamique de  $\varepsilon$  se produit alors à chaque fois que le nombre de solutions dans le front de Pareto dépasse la taille maximale  $N_A^{\text{max}}$  de l'archive  $A(t)$ .

$$\varepsilon_i = \frac{f_i^{\text{max}} - f_i^{\text{min}}}{n_{\text{div}_i} - (N_{A(t)} - N_A^{\text{max}})/\zeta} \quad (4.24)$$

$N_A$  est la taille de l'archive à l'itération  $t$  et  $\zeta$  est un coefficient positif supérieure à 2.

#### 4.10.5 Opérateurs de reproduction adaptés

Nous présentons ici, les méthodes de croisement et de mutation utilisées pour l'optimisation de la MPP et MPNP dans l'étape 4.10.3 de l'algorithme 7. Il est à rappeler que pour la MPP, les individus sont sous la forme de vecteurs réels. Chaque composante  $j$  (appelée aussi composant) du vecteur correspond à la fré-

quence de maintenance d'un sous-système  $j$  (faisant partie du système global). En revanche, pour la MPNP, les solutions sont sous la forme d'une matrice binaire. Pour chaque sous-système  $j$  correspond une ligne de la matrice qui retrace le planning de la maintenance du sous-système sur l'horizon temporel (cf. 4.3).

#### 4.10.5.1 Croisement de deux vecteurs réels : MPP

L'objectif est d'échanger du matériel génétique dans les couples d'individus sélectionnés au préalable. Après la formation des couples (faite de manière aléatoire, cf 4.10.3), les individus sont soumis ou non au croisement, selon une probabilité  $P_c$  définie préalablement dans l'algorithme. La probabilité  $P_c$  est généralement élevée car le croisement est le principal responsable de la création de nouveaux individus.

Soit  $P_1$  et  $P_2$  un couple d'individus appartenant respectivement à  $A(t)$  et  $P(t)$ . Nous vérifions s'il faut ou non procéder au croisement, par un tirage de probabilité  $P_c$ . Si c'est le cas, la combinaison du matériel génétique du couple  $(P_1, P_2)$  est faite comme suit dans les équations (4.25) et (4.26), afin d'engendrer deux types de descendance :

- Le premier type d'individus est voisin du parent qui a le plus de mérite, notamment  $P_1$  qui provient de  $A(t)$
- Le second type peut être proche de l'un ou de l'autre des parents

$$f_1 = p_1 - \alpha(p_2 - p_1) + (1 + 2\alpha)(p_2 - p_1)u(0,1)/2; \quad (4.25)$$

$$f_2 = p_1 - (1 - \beta)(p_2 - p_1) + (1 + 2\beta)(p_2 - p_1)u(0,1) \quad (4.26)$$

où  $u(0,1)$  est une variable distribuée uniformément dans  $[0,1]$  et  $\alpha$  et  $\beta$  des coefficients de multiplication.  $\alpha$  est choisi arbitrairement tel que  $\alpha \leq \frac{\sqrt{(3)}-1}{2} \simeq 0.366$  et  $\beta = 1 - \alpha$ . D'après [Dréo 2005], un  $\alpha \leq 0.366$  pour un croisement Blx- $\alpha = 0.63$  volumique; tel est presque notre cas; rime avec croisement contractant et voisin proche. En considérant l'équation (4.25) et compte tenu de la valeur de  $\alpha$ ,  $f_1$  a de fortes chances d'être plus proche de son parent  $P_1$ , qui a forcément plus de mérite que  $P_2$  vu qu'il appartient à  $A(t)$ . L'équation (4.25) se traduit finalement une forme d'approche de recherche locale.

En revanche avec un coefficient de multiplication  $\geq 0.5$  tel que  $\beta$  ( $\beta = 0.633$ ), l'équation (4.26) engendre des enfants qui peuvent être proches de l'un ou de l'autre des parents.

#### 4.10.5.2 Croisement de deux vecteurs binaires : MPNP

Un des challenges dans les algorithmes révolutionnaires est de concevoir un opérateur de croisement pertinent pour le problème en question, i.e. qui préserve les

bonnes caractéristiques des parents et perturbe les mauvaises. Cette approche de croisement représente une amélioration de l'approche présentée dans la section 4.8.2 et la figure 4.6 dans la mesure où elle traite mieux le problème de la taille des structures phénotypiques du couple d'individus participant au croisement. En effet, on exige qu'il n'y aura pas plus qu'une seule intervention de maintenance toutes les MTTF instants.

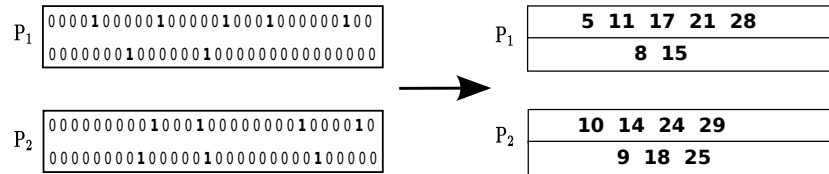


FIGURE 4.19 – Exemple de structures phénotypique non régulières de deux solutions

La figure 4.19 montre un exemple de deux parents P<sub>1</sub> et P<sub>2</sub> ayant des structures phénotypiques différentes et donc non adaptées (un ou les parents ne respectent pas la contrainte ci-dessus) pour faire directement le croisement. Cet écueil exige alors obligatoirement une étape intermédiaire pour uniformiser la taille des vecteurs qui forment la structure phénotypique.

Cette approche se déroule principalement en quatre étapes essentielles :

1. Découpage du génotype
2. Éclaircissement du génotype
3. Conversion génotype phénotype
4. Croisement des structures phénotypiques

**Première étape :** Comme dans l'approche 4.8.2 nous allons nous baser sur les données de fiabilité notamment le MTTF pour faire le découpage de chaque ligne de la matrice binaire (génotype) comme le montre la figure 4.20. Pour une ligne  $i$  donnée de la matrice binaire, il s'agit tout d'abord d'identifier le MTTF( $i$ ) correspondant et de la découper en segments de longueur MTTF( $i$ ) – 1, le dernier segment de la ligne étant  $\leq$  MTTF( $i$ ) – 1.

**Deuxième étape :** Elle consiste à repérer les segments binaires où le nombre de bit égale à 1 est  $\geq 2$  et ensuite à remplacer aléatoirement quelques uns de ces bits de façon à ce qu'il reste un seul bit égal à 1 dans un segment (voir figure 4.21).

**Troisième et quatrième étapes :** Il s'agit tout d'abord de convertir les matrices

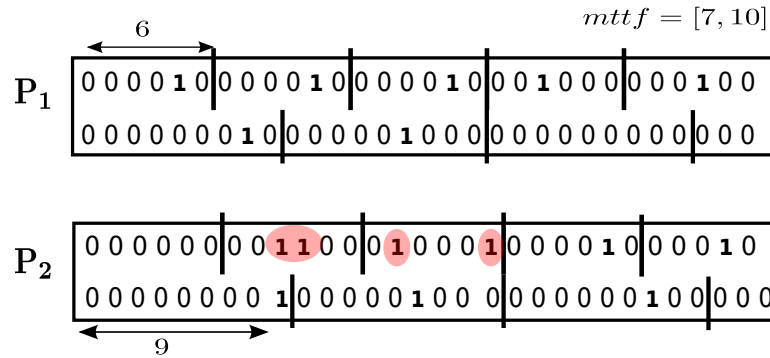


FIGURE 4.20 – Découpage du génotype

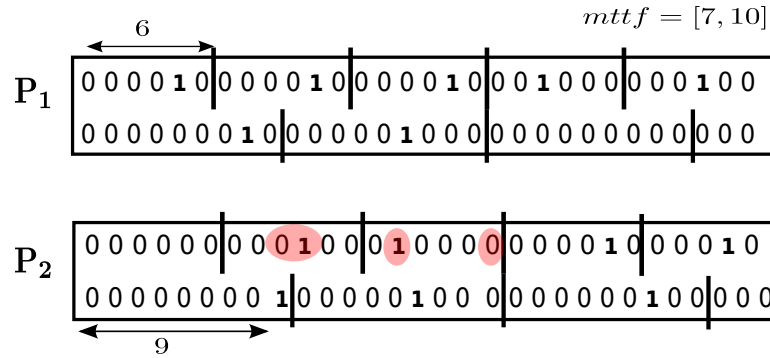


FIGURE 4.21 – Éclaircissement du génotype

binaires des parents en structures phénotypiques comme le montre la figure 4.22 suivante puis d'effectuer un croisement Blx- $\alpha$  volumique.

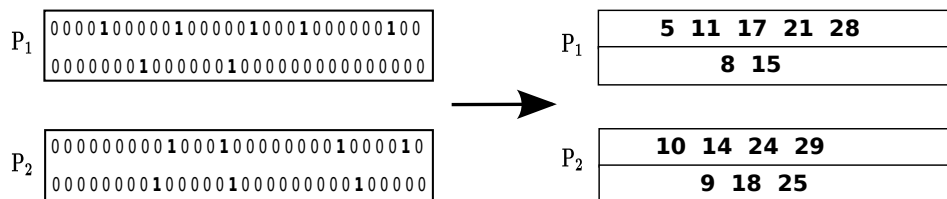


FIGURE 4.22 – Conversion du génotype en phénotype

#### 4.10.5.3 Mutation

La combinaison de la sélection et du croisement est normalement nécessaire pour assurer l'évolution de la population et la convergence de l'algorithme vers l'optimum. Mais, il arrive parfois que des informations essentielles contenues dans les gènes des individus disparaissent au cours des générations. Le rôle principal

de la mutation est de protéger les individus contre cette perte irrémédiable de matériel génétique.

Comme dans le croisement, la mutation est réalisée ou non en fonction d'une probabilité donnée  $P_m$  qui doit être faible. Pour le cas d'un individu à codage réel (MPP), la mutation consiste à ajouter un vecteur de perturbation  $\sigma$  à la portion de l'individu qui subit la mutation. Différemment, pour le cas des individus à codage binaire, l'opérateur de mutation consiste à compléter la valeur d'un bit du génotype (chromosome).

## **4.11 NSGA-II modifié pour l'optimisation de la maintenance préventive périodique**

Tout d'abord, il faut reproduire le NSGAII et l'adapter à notre problème d'optimisation de la maintenance. Le choix de cet algorithme de référence s'explique par sa réputation dans le domaine de l'optimisation multiobjectif. De plus, il nécessite peu de réglages de paramètres.

### **4.11.1 Description générale de l'algorithme**

L'algorithme NSGA-II utilise une procédure basée sur le critère de dominance de Goldberg pour attribuer un rang à chaque individu de la population. Cette technique de tri est fondée sur une classification des individus en fronts, à savoir que le rang de non-dominance attribué à un individu dépend de la frontière à laquelle il appartient. Ainsi, les individus non dominés appartiendront au premier front et auront un rang 0 et tous les individus non-dominés seulement par les individus du premier front forment le second front et front auront un rang 1 et ainsi de suite. La complexité algorithmique de cette procédure est de l'ordre  $O(v.N^2)$ , où  $N$  est la taille de la population et  $v$  le nombre d'objectifs.

Dans NSGA-II, chaque individu se voit également attribué une mesure de Distance de surpeuplement (DS). Cette dernière contrôle la diversité dans la population. Elle est définie pour les solutions du même rang. Une DS faible signifie que la région où se trouve la solution est dense, et inversement. Ainsi, entre deux solutions de même rang, on préférera celle ayant la DS la plus importante. Une moyenne de distance de surpeuplement élevée s'explique par une meilleure diversité dans la population. L'estimation de la DS d'une solution  $i$  repose sur le calcul de la distance moyenne sur chaque objectif, entre les deux solutions les plus proches situés de part et d'autre de  $i$ . Cet attribut est utilisé comme estimation de la taille du plus large *hypercube* n'incluant que le point  $i$  de la

population.

Le mécanisme d'évolution du NSGA-II repose sur une méthode de sélection par tournois. Si les individus participant au tournois appartiennent à des fronts différents, l'individu qui sera sélectionné est celui qui a le rang le plus bas, sinon, s'il y a égalité, c'est l'individu ayant la DS la plus élevée qui sera sélectionné. Cette technique permet d'avoir un front de Pareto bien étalé.

L'ensemble des individus sélectionnés forment ce qu'on appelle une population de parents notée  $P_t$  et de taille  $N$ . A chaque itération, une population d'enfants  $Q_t$  de taille  $N$  est engendrée à partir  $P_t$ . Les deux populations  $Q_t$  et  $P_t$  sont alors réunies dans une population  $R_t$  de taille  $2 \times N$ . Ensuite, la nouvelle population  $R_t$  est triée, selon le critère de dominance. Puis, une nouvelle population  $P_{t+1}$  de taille  $N$  est alors créée à partir des solutions des différents fronts de  $R_t$ . Les fronts de  $R_t$ , ne pouvant pas être contenus dans  $P_{t+1}$ , sont supprimés. Dans le cas où le nombre de solutions des différents fronts choisis pour intégrer  $P_{t+1}$  est supérieur à  $N$ , les individus ayant la DS la plus élevée sont choisis, tandis que les autres sont éliminés.

#### 4.11.2 Distance de surpeuplement dynamique

Dans les algorithmes évolutionnaires multiobjectifs, la diversité horizontale du front de partio est très importante. La diversité horizontale est souvent réalisé en enlevant l'excès d'individus dans l'ensemble de non-dominance lorsque le nombre de solutions non-dominées dépasse la taille de la population.

NSGA-II utilise la mesure de distance de surpeuplement comme indiqué dans l'équation (4.27) pour enlever le surplus d'individus. Les individus ayant de faibles valeurs de DS sont plus assujettis au processus d'élimination.

$$i_{DS} = \frac{1}{v} \sum_{k=1}^v |f_{i+1}^k - f_{i-1}^k| \quad (4.27)$$

$f_{i+1}^k$  est le  $k^{\text{ème}}$  objectif du  $i + 1^{\text{ème}}$  individu et  $f_{i-1}^k$  est le  $k^{\text{ème}}$  objectif du  $i - 1^{\text{ème}}$  individu après le tri de la population selon les mesures de DS. L'inconvénient majeur de la DS est que les solutions non dominées obtenues ne sont pas uniformément diversifiées. Pour surmonter ce problème, rendre la distance de surpeuplement dynamique (DSD) est une bonne solution récemment proposée dans [Luo 2008] et utilisé dans [Ramesh 2012] .

Dans cette approche, l'individu ayant la plus faible valeur de DSD est enlevé à chaque fois et la DSD est recalculée pour le reste des individus. Le DSD des individus est calculé comme suit :

$$i_{DSD} = \frac{i_{DS}}{\log(1/i_{DS})} \quad (4.28)$$



où  $i_{VDS}$  représente la variance des DSs des individus voisins du  $i^{\text{émè}}$  individu. Elle est calculée comme le montre l'équation (4.29) :

$$i_{VDS} = \frac{1}{v} \sum_{k=1}^v (|f_{i+1}^k - f_{i-1}^k| - i_{DS})^2 \quad (4.29)$$

$i_{VDS}$  donne des informations sur les variations de la DS à des objectifs différents. Supposons que pour une génération  $t$  la taille de l'ensemble des solutions non-dominées  $\mathcal{F}_t$  est  $M$ ;  $N$  est la taille de la population. Si  $M > N$ , alors l'approche d'élimination par DSD est utilisée pour enlever  $M - N$  individus de  $\mathcal{F}_t$ . Cette procédure peut se résumer par le pseudo code suivant :

---

**Algorithme 8** : Élimintaion de solutions non-dominées par l'approche de DSD

---

**Données** :  $\mathcal{F}_t$  : l'ensemble des solutions non-dominées à la génération  $t$

---

**tant que**  $|\mathcal{F}_t| > N$  **faire**

    Calcul-DSD( $\mathcal{F}_t$ );

    % *Trier  $\mathcal{F}_t$  selon les valeurs de  $i_{DSD}$*

    Tri-DSD( $\mathcal{F}_t$ );

    % *Eliminer la solution qui a la plus faible  $i_{DSD}$  dans  $\mathcal{F}_t$*

    Elimination( $\mathcal{F}_t$ );

**fin**

---

## 4.12 Conclusion

Ce chapitre se compose principale en deux parties. Nous avons proposé dans la première partie deux algorithmes mono-objectifs basées sur les algorithmes génétiques et le recuit simulé pour la résolution des problèmes d'optimisation de la maintenance préventive systématique. L'algorithme génétique proposé se décline en deux variantes. Les deux variantes se différencient principalement par le type du codage des solutions et de la nature des opérateurs de croisement et de diversification. Le recuit simulé se distingue par une stratégie de recherche de voisinage intelligente.

Dans la deuxième partie, deux algorithmes multiobjectifs ont été présentés. Dans le premier algorithme ev-MMOGA, la subdivision de l'espace des objectifs se voit être dynamique et les opérateurs de reproduction se déclinant en différentes variantes pour s'adapter le mieux au contexte de l'optimisation et à l'anatomie de la solution candidate. Le deuxième algorithme, le NSGAII-modifié se caractérise par une distance de surpeuplement dynamique, ce qui induit une nouvelle procédure d'élimination des solutions non-dominées. Dans le chapitre suivant, nous appliquons ces différents algorithmes sur un problème réel d'optimisation de la maintenance des rails.



# Optimisation de la maintenance préventive des rails

---

## Sommaire

5.1 Introduction . . . . .	133
5.2 Contexte de l'étude . . . . .	134
5.3 Modélisation de la dynamique de dégradation de la voie . . .	136
5.4 Optimisation mono-objectif de la MP des rails . . . . .	145
5.5 Optimisation multi-objectif : application à la MP des rails . .	152
5.6 conclusion . . . . .	159

---

## 5.1 Introduction

L'infrastructure est une composante essentielle du système ferroviaire, la voie ferrée en est l'acteur principal. Son dimensionnement autorise l'accroissement des capacités de transport, de charge et de vitesse, mais les contraintes environnementales et d'exploitation sont nombreuses.

La voie ferrée par elle-même réalise l'interface entre le sol et le matériel roulant dont il faut garantir les performances et la sécurité de circulation. Ainsi, les développements et progrès relatifs à la voie sont aussi importants que ceux du matériel roulant.

La voie, et notamment les rails subit les effets de la durée et du passage répété des trains avec nécessairement usures et dégradations. Il importe d'approfondir l'étude des procédures de maintenance de la voie compte tenu des conséquences économiques et sécuritaires qu'entraîne son dysfonctionnement. Le présent chapitre présente l'application de nos travaux sur un problème réel d'optimisation de la maintenance préventive du rail. La première partie est consacrée à la description et à la modélisation de la voie. Dans la deuxième partie les algorithmes d'optimisation développés dans le chapitre précédent sont mis en œuvre et leurs performances sont comparées.

## 5.2 Contexte de l'étude

La prévention des ruptures de rails est un point crucial pour conserver un niveau de sécurité et de disponibilité élevé du réseau de transport ferroviaire.

En effet, en plus de l'existence de facteurs environnementaux parfois contraignants (chaleur, froid, neige . . . ), les rails sont soumis à d'importantes contraintes mécaniques conjuguées (frottements, compressions/tensions au contact roue-rail), entraînant l'apparition de différents modes de dégradations tels que :

1. L'usure du profil transversal : elle est fonction du trafic (tonnage, nombre et vitesse des trains) et est la conséquence directe du frottement roue-rail, où des enlèvements de matière sont constatés soit sur la table de roulement soit sur la partie latérale surtout la zone de courbe serrée.
2. L'oxydation des rails qui se manifeste surtout dans les tunnels humides.
3. L'usure ondulatoire : ce phénomène identique à "la tôle ondulée" observé dans le domaine routier, est une usure périodique du profil en long du rail induit par des couplages complexes entre le matériel roulant (écartement des bogies, réglages des suspensions . . . ) et la voie (raideur vertical dureté des rails). Il se manifeste sous forme d'ondulations, plus ou moins régulières, dans la direction du roulement (figure 5.1).



FIGURE 5.1 – Usure ondulatoire du rail

4. Les défauts de fatigue : ils sont dûs à la modification des propriétés des matériaux (du rail) à l'application de cycles d'efforts, cycles dont la répétition peut conduire à la rupture [Bathias 2009]. On peut distinguer 3 phases dans le phénomène de fatigue : l'amorçage, la propagation des fissures, et la rupture brutale de la pièce. Pour ce qui concerne le rail, ces phénomènes de fatigue peuvent s'initier en surface (*squat head cheking*) ou en interne (fissure transverse).



FIGURE 5.2 – Fissure interne ayant conduit à la rupture

Ce dernier type de dégradation est le plus pénalisant car il aboutit à terme à la rupture systématique du rail. Les défauts de fatigue internes ou de surface apparaissent majoritairement au niveau des soudures aluminothermiques, points faibles de la voie. Toute anomalie, même légère, lors du soudage est susceptible d'entraîner une hétérogénéité du mélange (par exemple un préchauffage insuffisant). L'utilisation des moules de soudage peut également créer en surface des petites fissures qui constituent autant d'amorces de rupture. Notons cependant que l'amélioration des procédures de soudage a conduit à une forte diminution de ces inconvénients ces dernières années.

Le développement de ces fissures dépend principalement de la charge à l'essieu des matériels roulants circulant sur la voie et de leurs dynamiques verticale et horizontale. Plus la charge augmente, plus le phénomène de fatigue s'intensifie, accélérant la propagation des fissures.

L'inspection des rails, le suivi des défauts et les remplacements préventifs des coupons de rails sont donc les paramètres clefs pour le maintien des conditions de circulation.

Les premières inspections étaient réalisées visuellement au début du 19<sup>ème</sup> siècle. Depuis, les méthodes de diagnostic se sont développées, automatisées et spécialisées (techniques pour l'inspection de la géométrie de la voie, du profil du rail, des défauts internes du rails, ...). De nos jours, la plupart de ces opérations de surveillance se basent soit sur le passage d'un véhicule spécialisé effectuant des mesures ultrasonores et par des techniques à base de courants de Foucault soit sur la signalisation et en particulier les Circuits de Voie (CdV) qui permettent de détecter les ruptures de rails. Il est à noter que d'autres opérations de surveillance permettant de détecter en temps réel les ruptures, comme les parcours à pied et les signalements de conducteurs.

Des agents équipés de matériel de mesure sont également dépêchés sur zone pour surveiller l'évolution de certains défauts constatés. Suite à ce diagnostic, une action de maintenance sera possiblement décidée, s'il est jugé que l'état du rail est

critique.

L'enjeu est donc d'ajuster une fréquence de passage des systèmes d'inspection qui permette la détection des états critiques avant qu'ils n'évoluent en rupture de rail.

La réparation d'une rupture est une opération de maintenance curative visant l'insertion d'un coupon de rail neuf en lieu et place de la rupture. Cette méthode présente l'inconvénient de multiplier le nombre de soudures aluminothermiques, introduisant ainsi sur la voie des points singuliers à plus faible durée de vie. L'augmentation du nombre de soudures fragilise la voie, ce qui entraîne à termes le renouvellement complet de la voie.

### **5.3 Modélisation de la dynamique de dégradation de la voie**

Nous nous intéressons à présent, à la modélisation de la dégradation de la voie ferrée liée aux défauts internes de fatigue. Dans cette tâche nous nous sommes basés sur les résultats de l'étude effectuée par [Donat 2009] sur ce même problème réel.

#### **5.3.1 Représentation de la voie**

Pour représenter la voie relativement aux problèmes de dégradation, nous ferons l'hypothèse que celle-ci est principalement constituée de deux composants différent :

1. le rail standard dit "pleine barre"
2. les soudures aluminothermiques qui relient ces barres entre elles.



FIGURE 5.3 – Soudure aluminothermique

Comme les rails pleine barre sont, par nature, des composants continus et les soudures sont des points singuliers, la voie est discrétisée [Donat 2009] en consi-

dérant les hypothèses suivantes :

- Toute voie de longueur  $L$  est représentée sur ces 2 fils de rail par une succession de  $N_c$  coupons de longueur  $L_c$  où  $N_c = 2 \times L/L_c$ .
- Chaque coupon possède au plus une soudure aluminothermique.
- Pour une section de voie donnée, les coupons sont supposés indépendants possédant le même processus de dégradation.
- Les soudures sont supposées indépendantes et possèdent le même processus de dégradation.

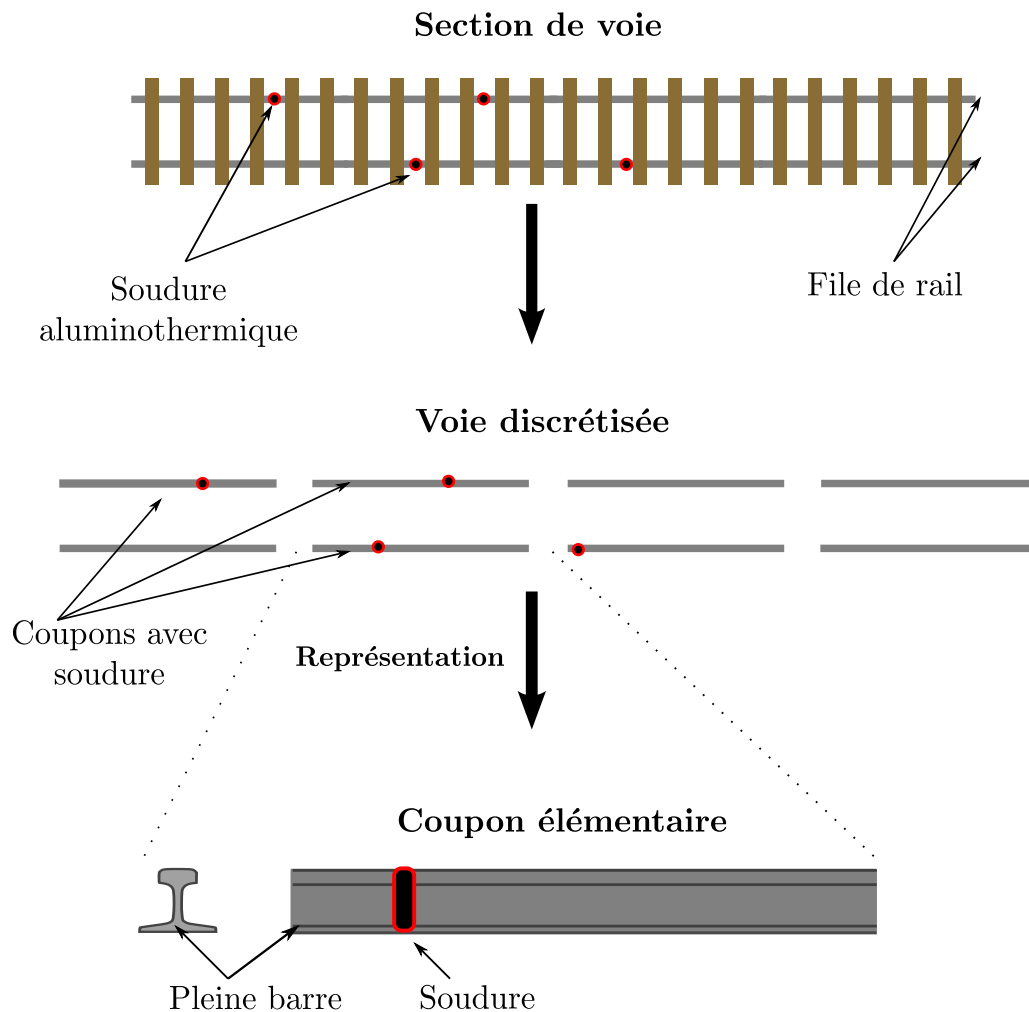


FIGURE 5.4 – Modélisation de la voie

D'après la figure 5.4 qui résume la procédure de modélisation de la dégradation, l'étude d'un couple élémentaire composé d' :

- un coupon de rail pleine barre de taille  $L_c$
- une soudure aluminothermique dont la présence sera gérée statistiquement.

### 5.3.2 Modèle de dégradation du coupon

Il s'agit à présent de procéder à la modélisation de la dégradation du coupon élémentaire vis à vis des problèmes de fatigue conduisant à la rupture (figure 5.4). Les composants (pleine barre + soudure) sont initialement sans défauts (N). Selon les normes en vigueur sur les réseaux urbains et interurbains français, les défauts de fatigue de type fissure transverse sont classifiés par rapport à leur taille, et par conséquent au degré d'urgence de leur élimination. Le classement, par gravité croissante, est le suivant :

- N : pas de défaut détecté ;
- O : défaut en observation ;
- X1 : défaut à surveiller de près ;
- X2 : défaut à réparer avec délai d'un mois maximum ;
- S : défaut à réparer d'urgence.

D'après les référentiels de maintenance, la classe S regroupe les ruptures, les défauts internes majeurs et les fissures visibles ayant atteint un stade critique. En revanche, pour des raisons de sécurité, seule la détection d'une rupture entraîne une réparation immédiate quelles qu'en soient les conséquences pour l'exploitation. Les autres défauts critiques sont généralement traités immédiatement, mais de manière palliative sans interrompre l'exploitation, puis de manière curative sous un certain délai.

Pour limiter la complexité algorithmique de l'approche et sachant qu'on s'intéresse, dans cette étude, à la prévention des ruptures des rails sur un horizon temporel large de l'ordre de 30 ans avec un pas de temps ou, quelques unes de classes de défauts sont fusionnées, pour former un ensemble d'états de dégradation d'un coupon élémentaire comme suit :

1. N : pas de défaut ;
2. OX1 : défaut mineur ;
3. X2S : défaut critique ;
4. SRup : rupture.

Dans cette application, les Modèles Graphiques de Durée (MGD, cf. chapitre 1, section 1.4.2) sont utilisés pour simuler la dynamique de dégradation des composants. Les lois de dégradation prennent en compte les paramètres contextuels de l'étude tellocalisation que le nombre de coupons élémentaire constituant une portion de voie et le facteur localisation. Ces variables de contexte sont indépendantes du temps et sont supportées par les MGD. Compte tenu des lois de la physique en jeux dans les problèmes de fissuration, il est fait l'hypothèse que les processus de dégradation ne sont pas auto-réparables et que l'état "rupture" est absorbant.



L'état global du coupon élémentaire est défini comme étant égal à l'état le plus critique entre celui de la pleine barre et celui de la soudure.

De manière explicite, nous posons  $X_t^S$  la variable représentant l'état du coupon élémentaire à l'instant  $t$ . Son éventuelle indisponibilité implique certaines pénalités économiques qui se chiffrent en termes de coût. La figure 5.5 montre le MGD associé au processus de dégradation d'un coupon élémentaire. Le modèle global contient deux sous-modèles correspondant aux deux composants pleine barre et soudure.

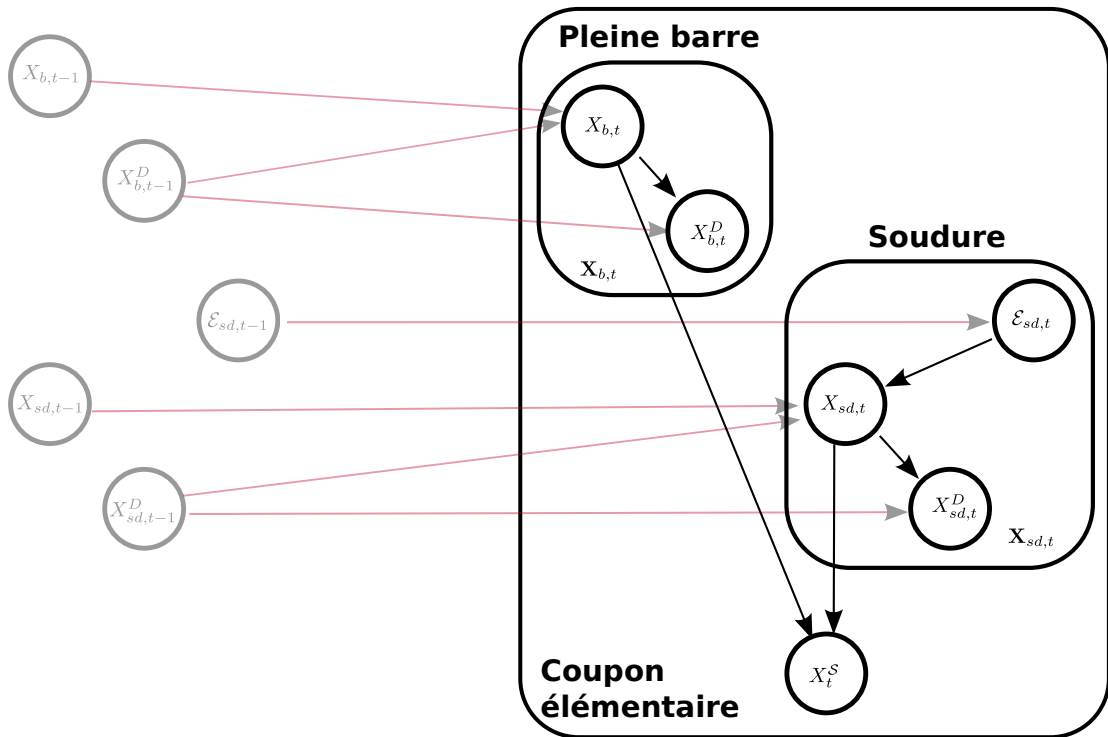


FIGURE 5.5 – Modèle de transition associé à la dégradation d'un coupon élémentaire. Les arcs en rouge indiquent une dépendance temporelle. Les arcs en noir indiquent les dépendances entre les variables dans une tranche temporelle.

### 5.3.2.1 Pleine barre

Le vecteur aléatoire  $\mathbf{X}_{b,t} = (X_{b,t}, X_{b,t}^D)$  représente les caractéristiques de la pleine barre à l'instant  $t$ .

- La variable aléatoire  $X_{b,t}$  à valeurs dans l'ensemble d'états  $\mathbb{X} = \{N, OX1, X2S, SRup\}$  désigne son niveau de dégradation.
- La variable aléatoire  $X_{b,t}^D$  à valeurs dans  $\mathbb{X}^D = \{1, \dots, T_S\}$  désigne, à l'instant  $t$ , le temps restant avant le prochain changement d'état;  $T_S$  correspond au temps

de séjour maximal dans chaque état.

#### Distribution probabiliste initiale

Le processus de dégradation naturelle d'une pleine barre est caractérisé par la distribution de son état à l'instant initial, notée  $E_{b,1}$ .

Rappelons que la loi  $E_{b,1}$  constitue un des paramètres descriptifs du modèle puisqu'elle permet de spécifier l'état initial des pleines barres sur la voie considérée.

En général, nous nous plaçons dans le cas où la voie est initialement neuve en affectant une probabilité égale à 1 à l'état N et 0 aux autres états :

$$P(X_{b,1}) = E_{b,1} \text{ tel que } P(X_{b,1} = N) = 1 \quad (5.1)$$

#### Loi probabiliste de transition

La Loi de Probabilité Conditionnelle (LPC) de transition notée  $E_{b,t}$  est détaillée dans le tableau 5.1. La LPC des temps de séjour quant à elle est notée  $F_{b,t}$ .

		$E_{b,t}$			
		$X_{pb,t}$			
$X_{pb,t-1}$		N	OX1	X2S	SRup
N		$\alpha$	$1 - \alpha$	0	0
OX1		0	$\beta$	$1 - \beta$	0
X2S		0	0	$\gamma$	$1 - \gamma$
SRup		0	0	0	1

TABLE 5.1 – Loi de probabilité conditionnelle de transition de la pleine barre sans maintenance

#### 5.3.2.2 Soudure

Le vecteur aléatoire  $\mathbf{X}_{sd,t} = (\mathcal{E}_{sd,t}, X_{sd,t}, X_{sd,t}^D)$  représente les caractéristiques de la soudure à l'instant  $t$ . La signification des variables aléatoires  $X_{sd,t}$  et  $X_{sd,t}^D$  est similaire à celle pour la pleine barre.

La variable stochastique  $\mathcal{E}_{sd,t}$ , à valeurs dans l'ensemble {présente, absente}, est une variable binaire indiquant si le coupon élémentaire présente ou non une soudure. L'absence de soudure est modélisée par une soudure neuve (i.e si  $\mathcal{E}_{sd,t}$  = absente alors  $P(X_{sd,t} = N) = 1$ ). En revanche, sa présence ne peut être possible que par le biais d'une intervention de maintenance corrective extérieure.

Nous notons par  $F_{sd,t}$  la LPC des temps de séjour de la soudure. D'après les

descriptions précédentes, la loi de transition naturelle de la soudure vérifie  $E_{sd,t}$  qui est décrite dans le tableau suivant :

		$E_{sd,t}$			
		$X_{sd,t}$			
$X_{sd,t-1}$	$\mathcal{E}_{sd,t}$	N	OX1	X2S	SRup
N	absente	1	0	0	0
OX1	absente	1	0	0	0
X2S	absente	1	0	0	0
SRup	absente	1	0	0	0
N	présente	$1 - \alpha$	$\alpha$	0	0
OX1	présente	0	$1 - \beta$	$\beta$	0
X2S	présente	0	0	$1 - \gamma$	$\gamma$
SRup	présente	0	0	0	1

TABLE 5.2 – Loi de probabilité conditionnelle de transition de la soudure sans maintenance

### 5.3.3 Politiques de maintenance modélisées

La décision de maintenance succède aux étapes de surveillance et de diagnostic comme le précise la figure 2.2 du chapitre 2. Dans cette section, nous présentons d'abord les méthodes de diagnostic utilisées pour l'inspection des voies dans le cadre de cette étude, ensuite la stratégie de sélection des actions de maintenance.

#### 5.3.3.1 Les outils de diagnostic

La politique de diagnostic consiste à déclencher une alerte dès qu'une des méthodes de surveillance détecte un défaut de fatigue sur une pleine barre ou une soudure. En pratique, les deux principales techniques de surveillance utilisées sur le terrain sont la méthode par ultrasons et la méthode par circuit de voie.

- L'objectif du diagnostic par ultrasons est de détecter les fissures internes dans ces phases initiales. En pratique, cela se fait de manière périodique, au moyen d'une voiture de contrôle équipée de sondes ultrasoniques qui circule sur zone à vitesse faible ( $< 50\text{km}$ ). Les inhomogénéités internes du rail provoquent des échos du signal ultrason et conduisent ainsi à détecter des fissures internes.
- Le diagnostic par Circuits de Voie (CdV), quant à lui, permet de détecter les ruptures de rails. A la base les CdV sont destinés permettent de détecter de façon automatique et continue la présence d'un véhicule sur une portion de voie donnée. Ils commandent ainsi l'activation des feux de signalisation placés

en bord de voie afin d'indiquer aux agents de conduite les autorisations de franchissement et les arrêts obligatoires. Ils fonctionnent grâce à l'introduction d'un courant dans les rails qui se trouve court-circuité par les essieux des véhicules et qui ne parvient plus aux récepteurs placés à la voie à intervalle régulier. L'absence de courant est ainsi interprétée comme le signe d'une présence d'un train sur la portion considérée (le "canton").

Mais lorsqu'il n'y a pas de véhicule sur la section de voie, les CdV sont également capables de repérer les rails cassés en observant en continu la disparition du courant de CdV. Hormis dans le cas de défaillances du CdV, cette diminution du courant est le signe d'une rupture de rail. Tout comme pour la technique par ultrasons, chaque diagnostic est confirmé ou infirmé par une équipe de maintenance qui se déplace sur les lieux de l'alerte.

L'objectif des outils de surveillance est de réunir un signalement pour tout défaut sur le coupon élémentaire quels que soient sa nature et son emplacement. Dès qu'un défaut de rupture est détecté, une opération de maintenance est déclenchée, aboutissant au remplacement du coupon défectueux, que l'avarie ait lieu en pleine barre ou sur une soudure.

Pour tenir compte des opérations de diagnostic et de maintenance, nous reprenons le modèle de dégradation présenté en figure 5.5 mais en lui ajoutant les composantes de diagnostic et de maintenance comme le montre la figure 5.6. Les variables aléatoires  $(D_{us,t}^b, D_{us,t}^{sd}, \delta_{us,t})$  représentent le diagnostic ultrasonore associé au coupon élémentaire.  $D_{us,t}^b$  et  $D_{us,t}^{sd}$  désignent respectivement les résultats du diagnostic pour la pleine barre et la soudure. L'activation de l'inspection ultrasonore à l'instant  $t$  dépend de la valeur de la variable binaire  $\delta_{us,t}$ .

De manière analogue,  $(D_{cdv,t}^b, D_{cdv,t}^{sd}, \delta_{cdv,t})$  modélisent le diagnostic par circuit de voie du coupon élémentaire. Le résultat final de la politique de diagnostic est représenté par la variable  $D_t^S$ .

Ce type de méthode de diagnostic est effectué de manière quasi continue ce qui fait que sa variable d'activation  $\delta_{sign,t} = 1$ . Nous notons respectivement par  $D_{cdv,t}^b$  et  $D_{cdv,t}^{sd}$  les variable désignant le résultat du diagnostic par signaux de la pleine barre et de la soudure.

Les résultats fusionnés des différentes techniques de diagnostic forment le diagnostic final du système,  $D_t^S$ . La variable aléatoire  $D_t^S$  est à valeurs dans  $\mathbb{D}^S = \{ND, D\}$  où ND désigne une absence d'alarme et D correspond à la détection d'un défaut. Autrement dit, lorsqu'au moins un signalement D est observé à l'instant  $t$  sur un composant, le diagnostic final est  $D_t^S = D$ . Dans le cas contraire, le diagnostic final prend la valeur  $D_t^S = ND$ .

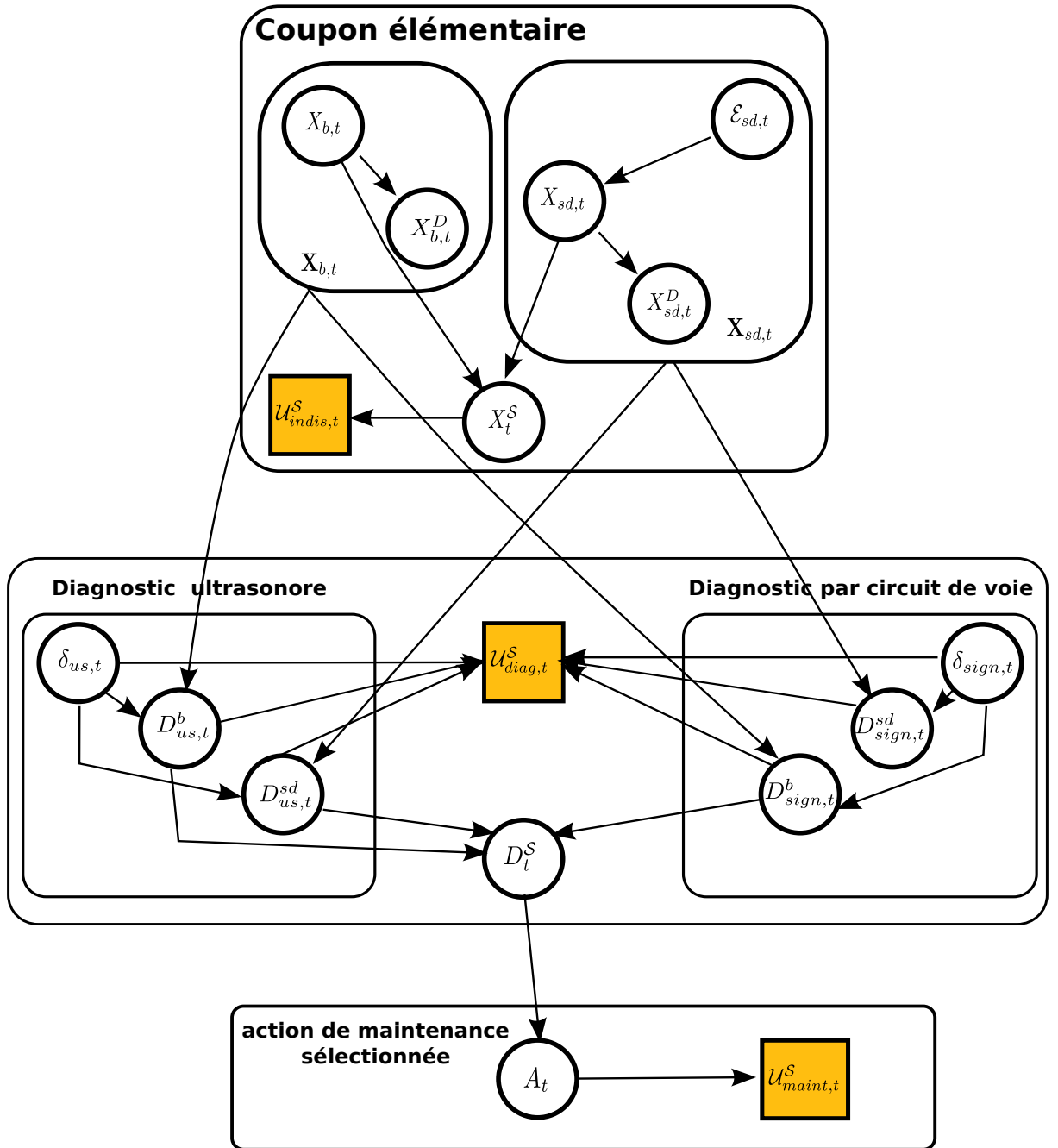


FIGURE 5.6 – Modèle graphique représentant le diagnostic et la maintenance de la voie à l'instant  $t$ .

### 5.3.3.2 Sélection de l'action de maintenance

Dans la figure 5.6, la politique de maintenance est modélisée par la variable aléatoire  $A_t$ . A chaque pas de temps, compte tenu du diagnostic final du coupon

élémentaire, une décision de maintenance est prise impliquant une intervention ou non sur le système. Quand l'équipe de maintenance intervient, c'est soit pour réparer le système (coupon élémentaire) de manière corrective en remplaçant la zone défectueuse par un coupon de rail sain ressoudé en voie, ou bien pour renouveler la portion de voie considérée. Il s'agit en fait, d'une politique de maintenance conditionnelle dont l'objectif est l'ajustement des différentes périodes d'inspection et des différents seuils de déclenchement des actions de maintenance. Les seuils critiques d'intervention sont définis par les différents états du système (voir section 5.3.2). L'optimisation des périodes d'inspection et de renouvellement de la voie est abordée dans section suivante.

Soit  $A_t$  la variable aléatoire désignant l'action de maintenance sélectionnée à l'instant  $t$ .  $A_t$  prend ses valeurs dans l'ensemble  $\mathbb{A} = \{A^0, A^c, A^r\}$ , où

- $A^0$  signifie qu'aucune action n'est réalisée.
- $A^c$  correspond à l'action curative sur le coupon élémentaire.
- $A^r$  correspond au renouvellement de la portion de voie considérée.

Pour prévenir les ruptures et limiter la dégradation des composants (propagation des fissures), les défauts mineurs OX1 sont surveillés en vue d'une réparation ultérieure tandis que les composants qui présentent des défauts critiques subissent une consolidation immédiate avant d'être réparés définitivement dans un délai plus court que l'unité temporelle. En revanche, une action de maintenance curative est immédiatement réalisée dès qu'une rupture est repérée. Dans le cas où un renouvellement de la voie est programmé à une date  $t$ , aucune autre action de maintenance ne peut être alors amorcée à cette même date.

L'état et les temps de séjour associés à la pleine barre et à la soudure à un instant  $t$  sont modifiés en fonction de l'action déclenchée au pas de temps  $t - 1$ . On suppose que le système "coupon élémentaire" est remis à neuf après une action de réparation ou de renouvellement. Les LPC de transition de la pleine barre et de la soudure seront définies alors de la même façon comme suit :

## 5.4 Optimisation mono-objectif de la MP des rails

En pratique, le renouvellement de la voie est effectué de manière périodique. Le but de l'optimisation est de chercher les fréquences de diagnostic et de renouvellement qui assurent une disponibilité optimale du système avec un moindre coût.

		$E_{b,t}$			
$X_{pb,t-1}$	$A_{t-1}$	$X_{b,t}$			
		N	OX1	X2S	SRup
N	$A^c$	1	0	0	0
OX1	$A^c$	1	0	0	0
X2S	$A^c$	1	0	0	0
SRup	$A^c$	1	0	0	0
N	$A^r$	1	0	0	0
OX1	$A^r$	1	0	0	0
X2S	$A^r$	1	0	0	0
SRup	$A^r$	1	0	0	0

TABLE 5.3 – Loi de probabilité conditionnelle de transition de la pleine barre avec maintenance

#### 5.4.1 Objectif de disponibilité

La productivité du processus industriel objet de cette étude se traduit par une meilleure disponibilité et qualité de service de la voie. La disponibilité passe principalement par la réduction des temps d'interventions pour maintenance que celle-ci soit de type réparation ou renouvellement total de la voie.

Cependant, nous considérons que le temps accordé à la maintenance est inférieur au pas de temps de la simulation car, les opérations de maintenance se déroulent principalement en dehors des périodes de service de la voie. En plus, des fausses alarmes déclenchées par les CdV, seule la présence d'une rupture sur le coupon élémentaire provoque son indisponibilité. Cette dernière pénalise le rendement et la qualité du service voyageur engendrant par conséquent des pertes économiques.

Nous quantifions la perte entraînée par l'indisponibilité du coupon élémentaire à un instant  $t$  comme suit :

$$U_{\text{indis},t}^S = u_t^S P(X_t^S = \text{SRup}) \quad (5.2)$$

On désigne par  $u_t^S$  le potentiel d'utilité (pénalité financière) associé à l'apparition d'une rupture sur le coupon élémentaire considéré l'instant  $t$ .  $P(X_t^S = \text{SRup})$  correspond à la probabilité qu'une rupture apparaisse sur le coupon élémentaire à l'instant  $t$ .

#### 5.4.2 Objectif budgétaire

L'amélioration de l'objectif de disponibilité requiert que les intervalles séparant les opérations de diagnostic-maintenance soient rapprochées. Néanmoins, ceci

peut entraîner un surcoût qui peut contrebalancer le gain atteint sur la disponibilité. C'est pourquoi, il est impératif de pouvoir mesurer de manière détaillée et précise les dépenses consacrées au diagnostic et aux interventions de maintenance. Nous procédons d'abord par l'évaluation du coût économique requis pour le déroulement du diagnostic.

Soit  $u_{\text{diag},t}^S$  le coût global associé au diagnostic du système à un instant  $t$ . Celui-ci inclut le coût de fonctionnement des systèmes de surveillance et le coût de validation du diagnostic, induit par le déplacement de l'équipe de confirmateurs chargée de confirmer le signalement émis par les équipements de surveillance.

Notons par :

- $u_{\text{us}}^{\text{surv}}$  : coût de l'inspection par ultrasons du coupon élémentaire ;
- $u_{\text{cdv}}^{\text{surv}}$  : coût de la surveillance par CdV par unité de temps de simulation ;
- $u_{\text{us}}^{\text{valid}}$  : coût de la validation d'un signalement ultrason ;
- $u_{\text{cdv}}^{\text{valid}}$  : coût de la validation d'un signalement lié au CdV.

L'estimation du coût des opérations de diagnostic pour un coupon élémentaire à l'instant  $t$  est effectuée de la façon suivante :

$$\begin{aligned} u_{\text{diag},t}^S = & u_{\text{us}}^{\text{surv}} \delta_{\text{us},t} + u_{\text{cdv}}^{\text{surv}} \\ & u_{\text{us}}^{\text{valid}} [P(D_{\text{us},t}^b \neq N) + P(D_{\text{us},t}^{\text{sd}} \neq N)] \delta_{\text{us},t} \\ & u_{\text{cdv}}^{\text{valid}} P(D_{\text{sign},t}^b \neq N) + P(D_{\text{cdv},t}^{\text{sd}} \neq N) \end{aligned} \quad (5.3)$$

Le budget consacré aux actions de maintenance doit couvrir les actions de réparation et de renouvellement. L'évaluation du coût associé aux opérations de maintenance sur un coupon élémentaire à l'instant  $t$  est défini par :

$$u_{\text{maint},t}^S = u_{\text{rep}} P(A_t^c = 1)(1 - \tau_{\text{ren}}) + u_{\text{ren}} \tau_{\text{ren}} \quad (5.4)$$

Où

- $u_{\text{rep}}$  est le coût d'une opération de maintenance corrective sur le coupon élémentaire ;
- $u_{\text{ren}}$  est le coût d'une remise à neuf du coupon élémentaire.
- $\tau_{\text{ren}}$  est une variable indicatrice à valeur dans  $\{0, 1\}$ . Elle est égale à 1 si l'instant courant  $t$  correspond à une date de renouvellement.

### 5.4.3 Paramètres de l'étude pour l'optimisation mono-objectif

Dans nos travaux, nous avons limité l'étude de la maintenance des rails à un horizon temporel  $T = 10$  ans. Ce choix est régit entre autres par la disponibilité des données de retour d'expériences qui servent à l'estimation des lois de temps de séjours du coupon élémentaire.



Dans une approche préventive des actions de maintenance les paramètres à optimiser sont la fréquence de surveillance par ultrasons  $f_{us}$  et la fréquence de renouvellement systématique de la voie  $f_{ren}$ . Nous limitons la variation de l'intervalle de surveillance entre 0 et 12 mois et celui du renouvellement entre 0 et 60 mois (5 ans). L'unité temporelle utilisée est le mois.

La simulation d'une politique de maintenance donnée consiste à inférer les successions de dégradations et de réparations d'un coupon élémentaire les plus proches possibles de la réalité terrain. Les réseaux bayésiens autorisent la compilation des retours d'expérience par le biais des tables de probabilités conditionnelles et les grandeurs stochastiques ainsi produites permettant effectivement de prendre en compte la variabilité des transitions d'états, les systèmes de diagnostic, voire la qualité des opérations de maintenance. Une moyenne sur plusieurs trajectoires d'état simulées permet le lissage des indicateurs de coûts pour l'évaluation d'une politique de maintenance donnée.

Pour la résolution du problème d'optimisation mono-objectif de la maintenance de notre cas d'étude réel, la fonction objectif est formée par l'agrégation des deux critères, présentés précédemment, formant ainsi la fonction d'utilité de la politique de maintenance déployée. La méthode utilisée pour l'agrégation des objectifs, est la méthode de pondération présentée dans le chapitre 3, section 3.4.1. La fonction objectif utilisée par les algorithmes d'optimisation mono-objectif s'écrit alors de la manière suivante :

$$\mathcal{F} = \frac{1}{T} \sum_{t=1}^T w_1 u_{indis,t}^S + w_2 [u_{diag,t}^S + u_{maint,t}^S] \quad (5.5)$$

Cette dernière évalue l'utilité moyenne mensuelle de la politique de maintenance préventive appliqué sur un coupon élémentaire de rail. Pendant cette étude l'hypothèse de  $w_1 = w_2 = 1$  est assumée. Afin de vérifier les résultats des algorithmes développés pour ce cas d'étude et présentés plus loin, une évolution exhaustive de  $\mathcal{F}$  en fonction des paramètres  $f_{us}$  et  $f_{ren}$  est donnée dans la figure 5.7. A partir de ce graphe, nous observons que l'optimum qui correspond à  $\mathcal{F} = 434830$  se situe à  $f_{us} = 2$  et  $f_{ren} = 35$ .

Nous traçons aussi, dans les figures 5.9 et 5.8 respectivement le coût moyen de l'indisponibilité et du diagnostic en fonction des périodes d'inspection et de renouvellement  $f_{us}$  et  $f_{ren}$ . Nous observons que lorsque la fréquence d'inspection  $f_{us}$  augmente, le coût du diagnostic diminue naturellement. En contrepartie l'indisponibilité du coupon élémentaire augmente et ceci d'autant plus que les périodes de renouvellement sont de plus en plus grandes.

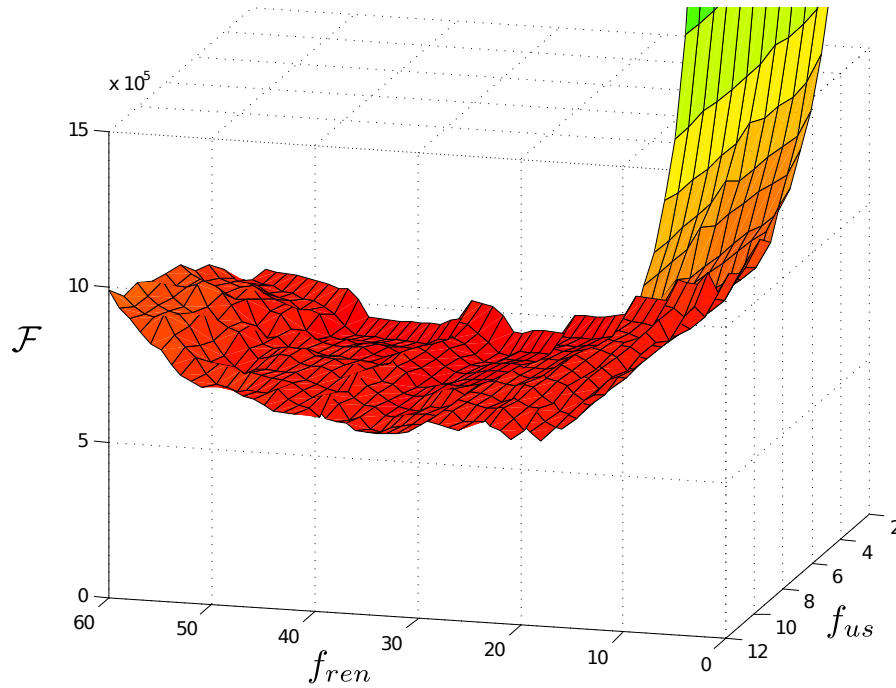


FIGURE 5.7 – Représentation de la fonction objectif  $\mathcal{F}(f_{us}, f_{ren})$  dans l'espace de solutions

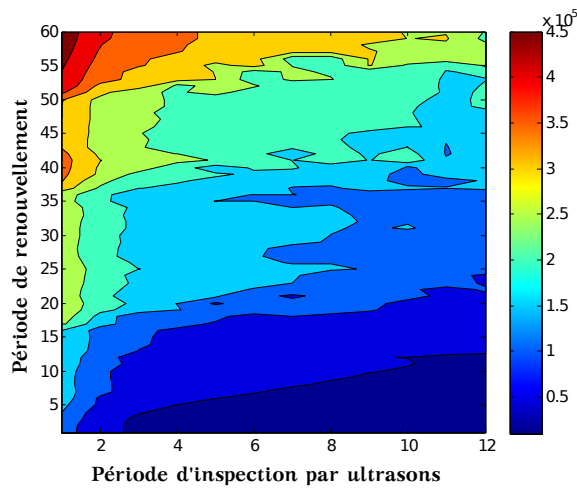


FIGURE 5.8 – Évolution du coût moyen mensuel lié à la maintenance du coupon élémentaire en fonction des périodes d'inspection et de renouvellement  $f_{us}$  et  $f_{ren}$ .

#### 5.4.4 Algorithme génétique

Nous avons appliqué l'algorithme AG-MPP détaillé dans 4.7.2 afin de rechercher les périodes d'inspections par ultrasons et de renouvellement optimales identifiées

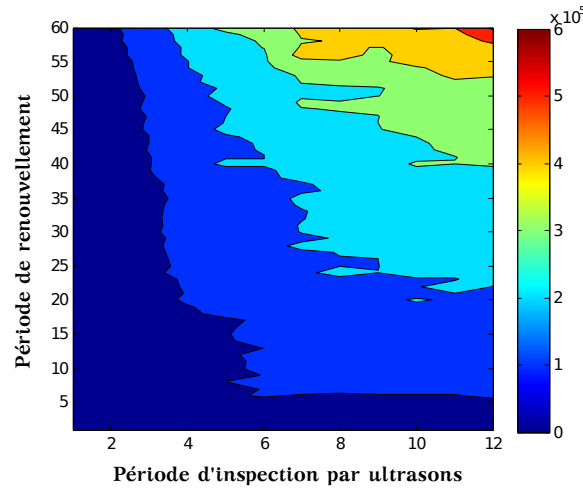


FIGURE 5.9 – Évolution du coût moyen mensuel lié à l'indisponibilité du coupon élémentaire en fonction des périodes d'inspection et de renouvellement  $f_{us}$  et  $f_{ren}$ .

au paragraphe précédent.

Il est important de signaler que le temps d'évaluation de la fonction objectif est relativement important (dizaines de secondes). De ce fait, une population très volumineuse de l'algorithme génétique risque de faire exploser le temps de calcul. L'issue est alors de débiter l'algorithme avec une population de taille petite, 10 individus/population dans notre cas; et puis d'opérer intelligemment des opérations sur les individus, comme le stipule la section 4.7.2, afin que l'algorithme converge le plus rapidement et le plus efficacement possible. L'efficacité de l'AG-MPP pour ce cas d'étude d'optimisation de la MP a été vérifiée au moyen d'une étude statistique. Pour ce faire, l'exécution du AG-MPP a été lancée une centaine de fois. Pour chaque simulation, le nombre d'évaluations total de la fonction objectif jusqu'à la convergence de l'algorithme a été enregistré. L'algorithme s'arrête soit aussitôt qu'il rencontre la solution optimale, soit après un nombre prédéterminé d'évaluations de la fonction objectif.

En examinant cet histogramme, il s'avère qu'une grande majorité des simulations convergent assez rapidement, avec parfois même, moins de 50 évaluations de fonction objectif. Ceci est considéré comme un résultat très encourageant en comparaison avec le nombre d'évaluations nécessaires (des centaines d'évaluations) pour une méthode de recherche exhaustive. Il faut rappeler qu'une caractéristique de l'AG-MPP est qu'il est évolutionnaire donc pouvant fournir des résultats approchés. Au vue de cette remarque, une analyse de qualité des résultats des simulations a été entreprise et illustrée dans la figure 5.11.

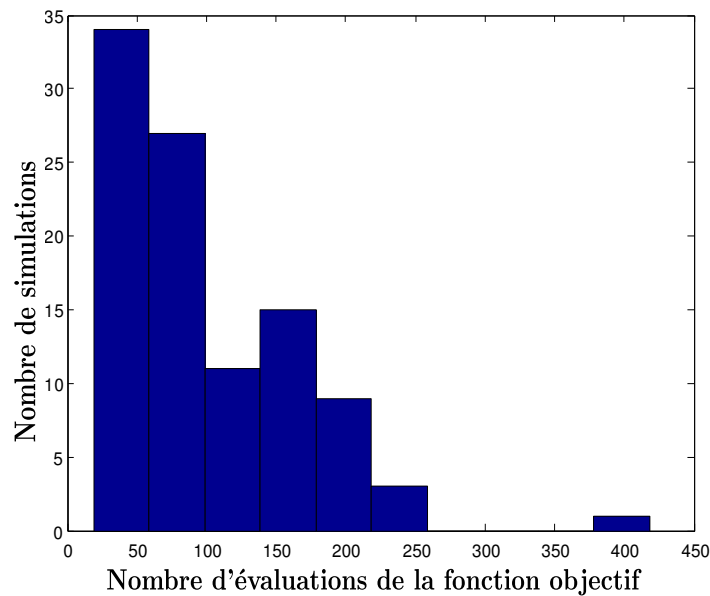


FIGURE 5.10 – Histogramme du nombre d'évaluations de la fonction objectif pour 100 simulations

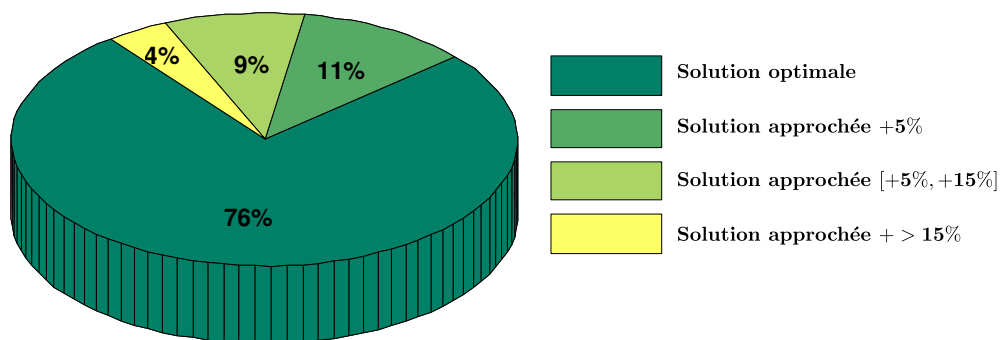


FIGURE 5.11 – Pourcentage des simulations ayant des solutions optimales et approchées.

#### 5.4.5 Recuit simulé modifié

Tout comme l'AG-MPP, l'algorithme du recuit simulé modifié (RSM), détaillé dans 4.9.1, a été appliqué pour l'optimisation de la MP du coupon élémentaire. Afin de comparer les performances de cet algorithme avec celui de l'AG-MPP, la même procédure de test a été effectuée. Il en résulte les figures 5.12 et 5.13.

L'historique du nombre d'évaluations, illustré dans la figure 5.12, montre que la vitesse de convergence du RSM est plus lente que celle de l'AG-MPP. Seulement un peu plus de 5% des simulations convergent avant la 50<sup>ème</sup> évaluation de la

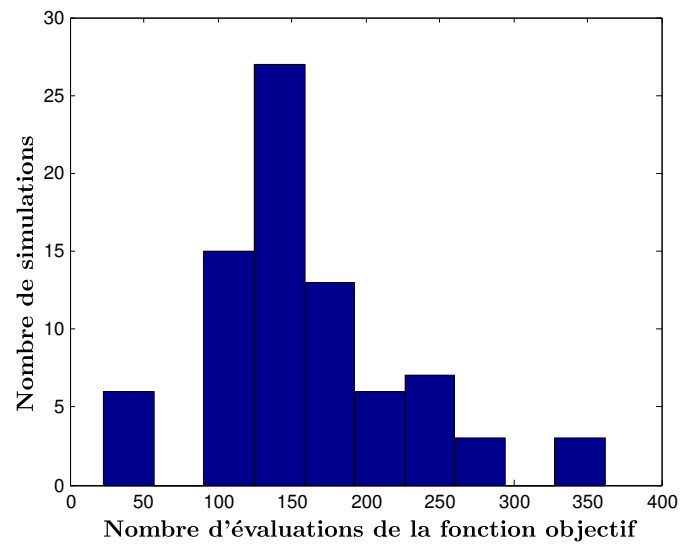


FIGURE 5.12 – Pourcentage du nombre de simulation du recuit simulé par rapport au nombre d'itérations

fonction objectif, contre 35% dans le cas de l'AG-MPP. La figure 5.13 affiche le

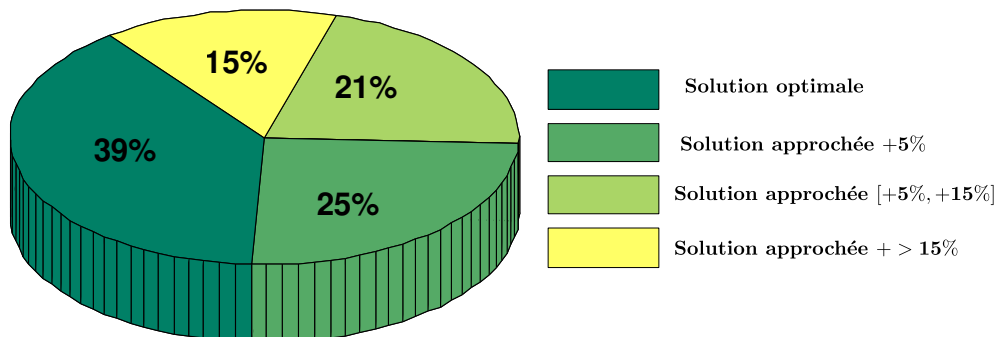


FIGURE 5.13 – Pourcentage des simulations ayant des résultats optimaux et approchés

pourcentage de simulations ayant des résultats optimaux et approchés. Moins de 40% des simulations convergent vers un résultat optimal, contre 76% dans le cas de l'AG-MPP. Si on agrège le taux de simulations optimales et approchées à +5%, le résultat pour le RSM s'améliore et passe à 64%.

Les simulations ayant des solutions approchées à +15% sont considérées comme non convergentes. A ce niveau, l'AG-MPP s'avère plus performant présentant seulement 4% de simulations non convergentes contre 15% pour le RSM.

### 5.4.6 Discussion

Nous constatons que les résultats obtenus par l'algorithme génétique sont nettement mieux par rapport à ceux du recuit simulé modifié. En effet, l'AG présente un temps de convergence inférieur que celui obtenu par le RSM en plus d'une meilleure qualité de solution. Cela s'explique principalement par la nature de l'AG qui est un algorithme à population et par l'efficacité de mécanismes de reproduction et de diversification de L'AG-MP. De ces constatations, nous pouvons dire que l'hybridation des algorithmes génétiques avec des méthodes de recherche locale améliore clairement les performances en terme de solution.

## 5.5 Optimisation multi-objectif : application à la MP des rails

Dans cette partie, le but est d'optimiser le fonctionnement de la maintenance préventive des rails en considérant séparément chaque critère influant sur le bon fonctionnement et sur les exigences pré-établies de la procédure de la maintenance.

### 5.5.1 Fonctions objectifs

Le problème d'optimisation multiobjectif de la maintenance de notre cas d'étude se compose principalement de deux critères d'optimisation. Les fonctions objectifs considérées sont donc :

- $\mathcal{F}_1$  : Coût de l'indisponibilité du coupon élémentaire par unité de temps.
- $\mathcal{F}_2$  : Somme des coûts des opérations de diagnostic et de réparation par unité de temps.

$$\mathcal{F} = \begin{cases} \mathcal{F}_1 = \frac{1}{T} \sum_{t=1}^T u_{\text{indis},t}^S \\ \mathcal{F}_2 = \frac{1}{T} \sum_{t=1}^T u_{\text{diag},t}^S + u_{\text{maint},t}^S \end{cases} \quad (5.6)$$

Comme, nous l'avons précisé dans 5.4.3, l'horizon temporel de l'étude est de 10 ans avec une discrétisation temporelle au mois. Nous pourrions discrétiser plus finement l'espace temporel pour arriver à une précision journalière ou hebdomadaire mais le déclenchement d'opérations de diagnostic et de réparation plus qu'une fois par mois est irréaliste financièrement parlant.

La figure 5.14 représente le paysage de l'espace des objectifs correspondant aux critères d'optimisation de la MP du coupon élémentaire.

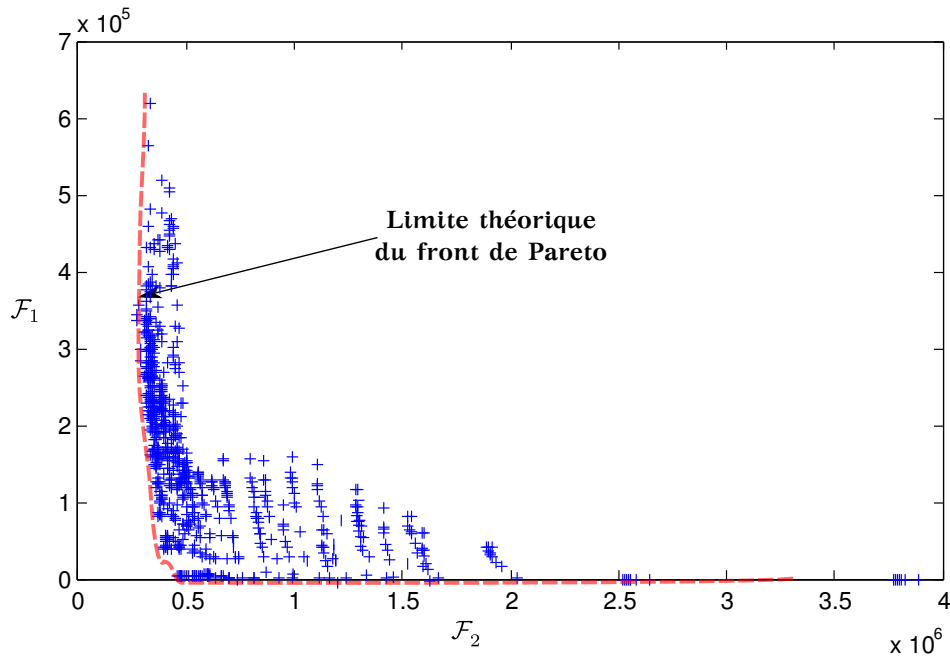


FIGURE 5.14 – Espace des objectifs

### 5.5.2 NSGA-II modifié

L'algorithme NSGA-II modifié détaillé dans la section 4.11.1 a été utilisé pour chercher les fréquences d'inspections et de renouvellement optimales, appliquées au coupon élémentaire. Le but est de trouver l'ensemble des solutions Pareto optimale garantissant un compromis entre les objectifs indisponibilité et coût de maintenance.

L'algorithme a été exécuté avec une population composée de 100 individus et avec un nombre d'itérations maximale égale à 100. Après seulement 5 itérations, nous remarquons que le front de Pareto approché obtenu à ce stade de l'algorithme n'est pas très loin du front de Pareto théorique optimal.

Pour voir la manière dont le front de Pareto évolue dans l'algorithme, nous l'avons tracé dans l'espace des objectifs. Le front de Pareto résultant après 50 et 100 itérations de la même exécution de l'algorithme est représenté respectivement dans la figure 5.15 et 5.16. Les points en rouge correspondent au front de Pareto optimal obtenu avec une population composée de 100 individus.

On constate qu'après la 50<sup>ème</sup> itération le front de Pareto est mieux tracé et plus large. L'amélioration apportée par les itérations suivantes de l'algorithme consiste plus à une meilleur distribution des solutions sur le front comme le montre la figure 5.16.

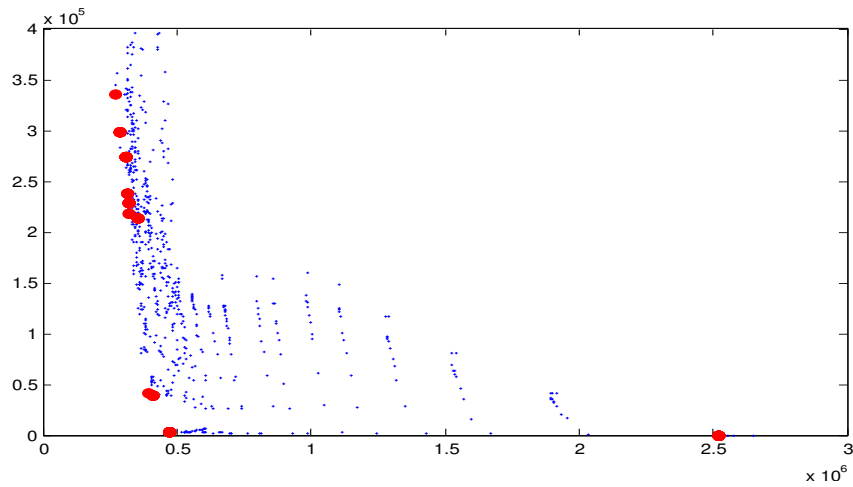


FIGURE 5.15 – Front de Pareto après 50 itérations de l'algorithme.

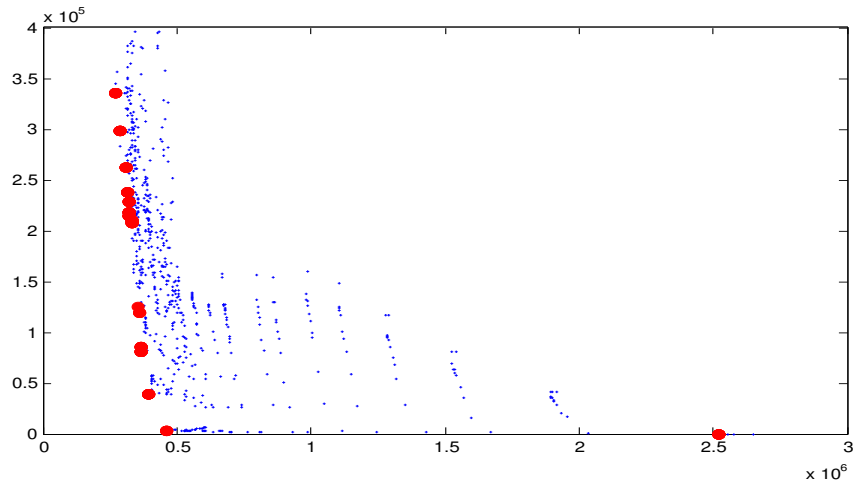


FIGURE 5.16 – Front de Pareto après 100 itérations de l'algorithme.

### 5.5.3 Ev-MMOGA

L'algorithme ev-MMOGA recherche les périodes optimales d'inspection et de renouvellement de la voie. Cet algorithme, détaillé dans la section 4.10, a été exécuté plusieurs fois en variant à chaque fois le nombre d'individus par population (20 / 50/ 100 individus par population) et le critère d'arrêt de l'algorithme (nombre maximal d'itérations).

Il a été constaté qu'une taille minimale de population correspondant à 20 individus est suffisante pour que l'algorithme converge vers un ensemble de solutions Pareto optimales. Aussi, pour toutes les exécutions réussies, l'algorithme converge vers un front de Pareto optimal bien avant la 100<sup>ème</sup> itération.



De manière générale, l'algorithme se rapproche vers un même front de Pareto optimal assez rapidement; un résultat approché encourageant est même visible, le plus souvent, des les premières générations comme le montre la figure 5.17. Le

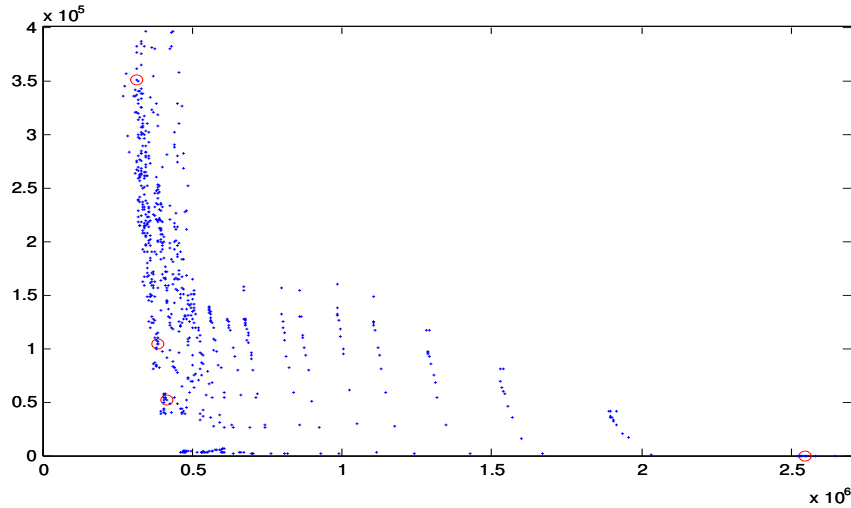


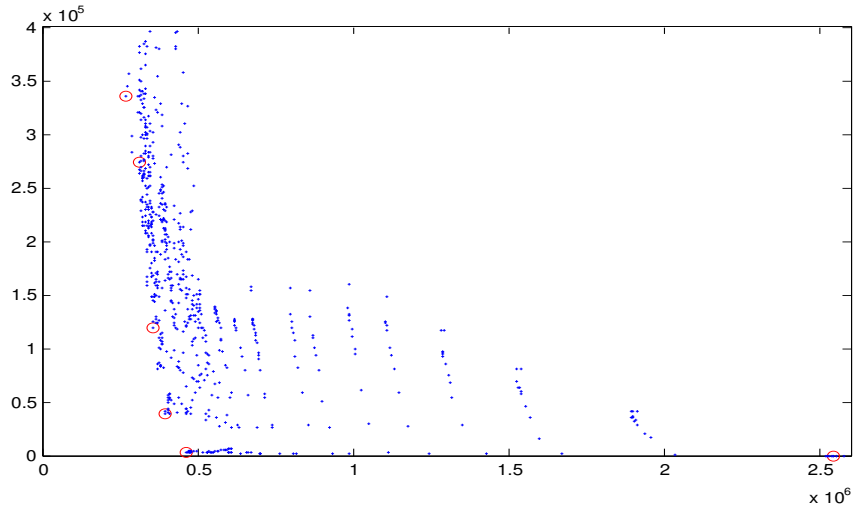
FIGURE 5.17 – Front de Pareto approché fourni par la 5<sup>ème</sup> génération d'une population composée de 50 individus. Les petits cercles en rouge délimitent le front de Pareto; les points en bleu représentent des solutions faisables de l'espace des objectifs

front de Pareto dans la figure 5.17 est délimité par des petits cercles rouges, il est obtenu après la 5<sup>ème</sup> itération de l'algorithme. Nous observons alors, que ce front de Pareto se rapproche étroitement du front de Pareto théorique optimal tracé dans la figure 5.14.

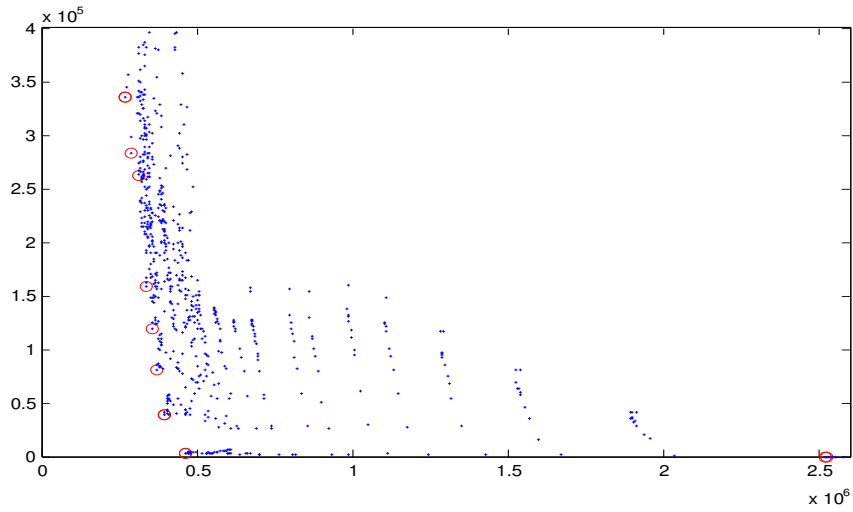
Pour la même exécution de l'algorithme, les figures suivantes 5.18a et 5.18b révèlent les fronts de Pareto obtenus après les 50<sup>ème</sup> et 100<sup>ème</sup> itérations. En comparant les fronts de Pareto des figures 5.18a et 5.18b, nous notons que l'ensemble des solutions de, l'archive s'est enrichi avec de nouvelles solutions sans pour autant encombrer les solutions déjà existantes. Les solutions du front de Pareto final (après la 100<sup>ème</sup> itération) sont ainsi bien distribuées sur le front de Pareto.

#### 5.5.4 Comparaisons des algorithmes

Dans le cas multiobjectif, les algorithmes ne peuvent plus être jugés sur la qualité de la solution fournie. De ce fait, de nombreuses métriques ont été définies, afin de juger de la qualité d'un front de Pareto [Okabe 2003]. Dans la suite de cette sous section, nous présentons les trois métriques utilisées pour évaluer les performances du NSGA-II modifié et du ev-MMOGA, appliqués à la MP des rails.



(a) après 50 itérations de l'algorithme



(b) après 100 itérations de l'algorithme

FIGURE 5.18 – Espace des objectifs : les points en rouge correspondent au front de pareto optimal obtenu avec une population composée de 100 individus.

### Covariance

Soit  $A$  et  $A'$  deux fronts de Pareto donnés par deux algorithmes différents  $(\mathcal{A}, \mathcal{A}')$ . La covariance  $C(A, A')$  mesure le ratio de solutions de l'archive  $A'$  qui sont dominées par des solutions de l'archive  $A$ .  $C(A, A')$  est définie par l'équation 5.7 :

$$C(A, A') = \frac{|\{u \in A' \mid \exists v \in A, v \text{ domine } u\}|}{|A'|} \quad (5.7)$$

où l'opérateur  $|\cdot|$  donne le cardinal de l'ensemble.

$C(A, A') = 1$  implique que tous les vecteurs contenus dans  $A'$  sont dominés par

ceux contenus  $A$ . En revanche,  $C(A, A') = 0$  signifie qu'aucun des vecteurs de  $A'$  n'est dominé par un des vecteurs de  $A$ .

Ainsi, plus le couple  $(C(A, A'), C(A', A))$  est proche de  $(1, 0)$ , plus l'algorithme  $A$  aura tendance à être meilleur que l'algorithme  $A'$ . Deux algorithmes ont des performances semblables si  $(C(A, A'), C(A', A))$  tend vers  $(0.5, 0.5)$ .

### Espacement

L'espacement **SP** du front de Pareto mesure la distribution des solutions stockées dans l'archive le long du front de Pareto. Soit  $A$  l'archive qui contient le front de Pareto. L'espacement  $S(A)$  du front de Pareto est défini par les équations 5.8 et 5.9 :

$$SP(A) = \sqrt{\frac{1}{|A|-1} \sum_{i=1}^{|A|} (d_i - d_{\text{moy}})^2} \quad (5.8)$$

$$d_i = \min_{x_j \in A \wedge x_j \neq x_i} \left( \sum_{\ell=1}^k |\mathcal{F}_\ell(x_i) - \mathcal{F}_\ell(x_j)| \right) \quad (5.9)$$

$|A|$  est le nombre de solutions stockées dans l'archive  $A$ ,  $k$  est le nombre de fonctions objectifs et  $d_{\text{moy}}$  est la distance moyenne de  $d_i$ . Une grande valeur de  $SP(A)$  démontre que les solutions sont bien réparties le long du front de Pareto.

### Largeur du front de Pareto

La largeur du front de Pareto proposée dans [Zitzler 1999b, Zitzler 2000] évalue la distance entre les solutions limites du front de Pareto. Elle est notée par **LP** est définie par l'équation 5.10 :

$$LP = \sqrt{\sum_{\ell=1}^k \left( \max_{i=1}^{|A|} (\mathcal{F}_\ell(x_i)) - \min_{i=1}^{|A|} (\mathcal{F}_\ell(x_i)) \right)} \quad (5.10)$$

### Résultats expérimentaux

Les résultats obtenus avec ev-MMOGA sont comparés à ceux obtenus par le NSGA-II modifié. Les deux algorithmes sont exécutés avec une population de taille 100 et un critère d'arrêt correspondant à 100 itérations.

Après plusieurs expérimentations en variant à chaque fois le critère d'arrêt, nous constatons que les deux algorithmes convergent bien avant la 100<sup>ème</sup> génération. Par conséquent, le front de Pareto après la 100<sup>ème</sup> génération correspond au résultat final de l'algorithme. Dans l'algorithme ev-MMOGA, la taille maximale de l'archive est bornée à 20.

- les couples moyens  $((C(A_{\text{NSGAII-modif}}, A_{\text{ev-MMOGA}}), (C(A_{\text{ev-MMOGA}}, A_{\text{NSGAII-modif}})))$ ;
- les mesures d'espacement moyennes de front de Pareto;
- les largeurs moyennes de front de Pareto

	Covariance	Espacement	Largeur du front de Pareto
NSGA-II modifié	0.0800	0.0582e+04	2.2742e+06
ev-MMOGA	0.4615	1.7037e+04	2.2774e+06

TABLE 5.4 – Tableau comparatif des algorithmes par rapport à la covariance, l’espacement et la largeur du front de Pareto

Le tableau 5.4 présente les résultats des algorithmes NSGA-II et ev-MMOGA sur 25 exécutions. D’après les résultats illustrés dans le tableau, il est clair qu’en terme de covariance, le NSGA-II modifié est meilleur que le ev-MMOGA. Par ailleurs, pour ces deux algorithmes, la taille maximale de l’archive externe est égale à 100 pour le NSGA-II et de 50 pour le ev-MMOGA. La comparaison de la distribution (espacement) **SP** des deux algorithmes quant à elle montre que l’ev-MMOGA est nettement meilleur. Ce résultat est dû entre autres à la subdivision dynamique de l’espace des objectifs. Ce mécanisme en plus de permettre de gérer la densité des solutions sur le front de Pareto, il permet d’ajuster les limites de front de Pareto s’adaptant ainsi aux caractéristiques du Problème. À tout prendre, en considérant aussi la critère de la largeur **LP**, qui est favorable pour l’ev-MMOGA, nous constatons que ce dernier assure une meilleure couverture du front de Pareto et semble être tout à fait adapté à notre problématique.

## 5.6 conclusion

Dans ce chapitre, nous avons présenté l’application de la méthodologie de résolution de l’optimisation de la maintenance d’un système, à un problème réel de maintenance de la voie ferroviaire. Les différentes étapes de cette méthodologie ont été décrites en commençant par la représentation et la modélisation de la dégradation de la voie puis intégrant l’évaluation de la maintenance et enfin l’optimisation des paramètres de la stratégie de maintenance déployée.

L’étude de la dégradation de la voie a été tout d’abord ramenée à l’étude d’un coupon élémentaire (système à deux composants : pleine barre et soudure) représentatif de la voie et se dégradant au cours du temps. Puis, en utilisant l’environnement de modélisation VirMaLab, la dégradation, les opérations de diagnostic et de maintenance du coupon élémentaire ont été modélisés par le formalisme des modèles graphiques de durée. Dans ce modèle des nœuds d’utilités sont introduits, ce qui permet de construire la fonction de coût permettant d’évaluer la politique de maintenance. L’optimisation a été par la suite abordée sous deux angles : mon-objectif et multiobjectif. Dans le cas mon-objectifs, les algorithmes AG-MP et RSM proposés dans ce travail ont été appliquées et puis

leurs performances comparées. Il a été constaté que l'AG-MP qui est un algorithme évolutionnaire à population, donne de meilleurs résultats que le RSM et est satisfaisant en termes de vitesse de convergence. Pour le multiobjectif, les algorithmes ev-MMOGA et NSGA-II modifié ont été aussi exécutés et ont donné un résultat encourageant. En termes de différents critères comparatifs, il s'est avéré que que les deux algorithmes ont des performances proches, mais l' ev-MMOGA se détache grâce à sa technique particulière de découpage de l'espace favorisant ainsi l'espacement des solutions et l'étalement du front de Pareto. Fort des résultats numériques fournis par l'ev-MMOGA et l'AG-MP, il s'est conclu que ces deux algorithmes s'adaptent très bien à notre problématique d'optimisation de la maintenance préventive systématique.



# Conclusion et perspectives

Dans cette thèse, nous nous sommes intéressés au développement d'une démarche de résolution générique d'un problème d'optimisation de la maintenance préventive des systèmes dynamiques multi-composants. Les travaux rapportés dans ce manuscrit abordent principalement deux grands axes de recherche. Le premier porte sur l'étude des processus de dégradation des systèmes multi-composants, multi-états et des politiques de maintenance. Ce premier axe ramène au deuxième qui se rapportent à sur la recherche des paramètres de configuration optimaux d'une politique de maintenance préventive donnée. Après avoir abordé ces questions dans les précédents chapitres, nous avons synthétisé les principales contributions de cette thèse et déceler les perspectives envisagées.

Après avoir passé en revue l'état de l'art des approches de modélisation des dégradations existantes dédiées aux systèmes industriels, l'approche de modélisation par modèles graphiques probabilistes a été retenue. Cette dernière est fondée sur le formalisme particulier des modèle graphiques de durée. Avec leur dynamique d'évolution complexe et leurs capacités de modélisation étendues, les modèles graphiques de durées s'adaptent tout à fait pour formaliser les relations de dépendance entre les différents composants d'un système complexe évoluant au cours du temps. De plus, ils ont le mérite de considérer des temps de séjour distribués selon des lois quelconques dans chacun des états du système. Cette propriété surmonte la limitation imposée par les chaînes de Markov, qui exigent obligatoirement une loi de temps de séjour de type exponentielle.

La modélisation de la dégradation est suivie par l'étape d'intégration des approches de maintenance dans le modèle et la définition d'une fonction générique pour l'évaluation des politiques de maintenance. Cette dernière permet de quantifier la pertinence d'une politique de maintenance donnée par rapport à une autre, sur la base d'un ou plusieurs critères. L'évaluation d'une stratégie de maintenance est guidée par un calcul d'utilité soulignant sa capacité à prendre en compte simultanément plusieurs contraintes (économiques, logistiques, de sûreté de fonctionnement ...). Il est à noter que cette évaluation est rendu possible grâce à la capacité du modèle à simuler un grand nombre de politiques de maintenance correctives et préventives.

Ainsi, sous réserve de disposer de la configuration optimale de la politique de maintenance préventive d'un système donné soumis à un environnement et des contraintes particulières, la maintenance préventive peut être nuisible et inefficace en générant des coûts économiques inutiles et beaucoup d'indisponibilités du système si le planning des interventions n'est pas bien réglé. Cet écueil, rend

l'application de méthodes d'optimisation visant à ajuster les paramètres de maintenance incontournable.

Cette optimisation s'effectue en agissant sur les paramètres de réglage de la politique de maintenance inclus dans le modèle.

Dans ce contexte particulier d'optimisation, les méthodes d'optimisation exactes s'avèrent inappropriées principalement à cause du temps de calcul qui augmente de manière exponentielle proportionnellement à la dimension du problème. De plus, ces méthodes exigent une connaissance portant sur la forme et la nature de l'objectif à optimiser ce qui n'est pas souvent possible. Ces limites invitent ainsi à s'orienter vers les approches d'optimisation approchées et notamment les métaheuristiques. Outre leur capacité à s'appliquer à un très large panel de problèmes, ces méthodes permettent d'aborder le problème d'optimisation sous un angle mono ou multi objectifs. Néanmoins, il faut noter que ces méthodes sacrifient l'exactitude de l'optimum et la complétude de la résolution pour gagner en efficacité et maîtriser le temps de calcul.

Un des enjeux de cette thèse était de proposer des algorithmes d'optimisation mono-critères et multicritères adaptés et efficaces pour la résolution de la problématique particulière de l'optimisation de la maintenance préventive. Nous avons de ce fait, développé des algorithmes évolutionnaires, mono-objectifs (AG-MP et RSM) et multiobjectif (ev-MMOGA et NSGA-II modifié) pour l'optimisation des paramètres de la maintenance préventive systématique (périodique, non-périodique). Pour développer ces métaheuristiques, plusieurs points étaient à définir et pour lesquels différents choix étaient possibles. Cette manière de fonctionner les rend ainsi plus génériques et en même temps plus faciles à adapter et à spécialiser.

L'algorithme génétique développé AG-MP se décline en deux variantes. La première décrit, l'AG-MPP traite l'optimisation de la maintenance préventive périodique. La deuxième, un algorithme génétique spécifique, adapté pour l'optimisation de la maintenance préventive non périodique. De part sa nature (algorithme à population) et sa dynamique d'évolution (opérateurs de reproduction et de diversification), l'AG-MP donne de meilleurs résultats que le RSM lorsqu'ils sont comparés avec des benchmarks d'optimisation et plus spécialement lorsqu'ils sont appliqués à un cas d'optimisation de la maintenance.

La première métaheuristique multiobjectif expérimentée a été le ev-MMOGA. Celui-ci a été adapté aux problèmes de la maintenance préventive systématique afin d'approcher les paramètres de réglage optimale d'une telle politique de maintenance. La seconde que nous avons adaptée est le NSGAII. Après les réglages et modifications nécessaires, cette métaheuristique a été proposée avec une nouvelle



procédure d'élimination des solutions non-dominées. Cette procédure, est dû à une distance de surpeuplement dynamique.

Enfin, nous avons procédé à l'application de la démarche à l'étude d'un cas réel industriel d'optimisation de la maintenance dans le domaine ferroviaire. Avec cette approche, nous avons pu souligner les spécificités de l'approche et sa capacité à modéliser le comportement du système, évaluer les stratégies de maintenance et à les optimiser.

Les perspectives de ce travail sont nombreuses, aussi bien en pratique qu'en théorie. Il est en effet possible d'appliquer cette méthodologie de l'optimisation de la maintenance à d'autres problèmes industriels tels que le diagnostic automobile, la maintenance de systèmes mécaniques dans le domaine aéronautique, certains problèmes ordonnancement des tâches dans les chaînes de production.

Les algorithmes peuvent être étendues aussi à la maintenance préventive conditionnelle. Dans ce cas, l'objectif serait d'optimiser des seuils maximaux de défaillances, au delà desquels différents types d'actions de maintenance pourraient être envisagées. Enfin, une perspective intéressante reste l'application des méta-heuristiques à l'optimisation en temps réel des paramètres de maintenance. Cette application revient à résoudre un problème d'optimisation dynamique.



# Bibliographie

- [Abdel-Hameed 1975] Mohamed Abdel-Hameed. *A Gamma Wear Process*. IEEE Transactions on Reliability, vol. R-24, no. 2, pages 152–153, Juin 1975.
- [Allaire 2005] G. Allaire. *Analyse numérique et optimisation : une introduction à la modélisation mathématique et à la simulation numérique*. Mathématiques appliquées. École polytechnique, 2005.
- [Allenson 1992] Robin Allenson. *Genetic Algorithms with Gender for Multi-function Optimisation*. 1992.
- [Arulampalam 2002] M. Sanjeev Arulampalam, Simon Maskell et Neil Gordon. *A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking*. IEEE Transactions on Signal Processing, vol. 50, pages 174–188, 2002.
- [Asmussen 2003] Soeren Asmussen. *Applied Probability and Queues (Stochastic Modelling and Applied Probability)*. Springer, 2nd édition, Mai 2003.
- [Ayadi 2011] Ines Ayadi, Laurent Bouillaut, Patrice Aknin et Patrick Siarry. *Optimal preventive maintenance schedules using specific genetic algorithms and probabilistic graphical model*. page 901â909, 2011.
- [Bagdonavicius 2001] Vilijandas Bagdonavicius et Mikhail S. Nikulin. *Estimation in Degradation Models with Explanatory Variables*. Lifetime Data Analysis, vol. 7, no. 1, pages 85–103, Mars 2001.
- [Bar-Noy 2004] A. Bar-Noy, V. Dreizin et B. Pattshamir. *Efficient algorithms for periodic scheduling*. Computer Networks, vol. 45, no. 2, pages 155–173, June 2004.
- [Barbour 1995] A. D. Barbour, O. Chryssaphinou et M. Roos. *Compound Poisson approximation in reliability theory*. IEEE Transactions on Reliability, vol. 44, no. 3, pages 398–402, Septembre 1995.
- [Barlow 1987] Richard E. Barlow et Frank Proschan. *Mathematical Theory of Reliability (Classics in Applied Mathematics)*. Society for Industrial Mathematics, Janvier 1987.
- [Bathias 2009] C. Bathias et A. Pineau. *Fatigue des matériaux et des structures : Tome 3, fatigue à haute température, effet des entailles polymères et élastomères, approche probabiliste, prévision croissance des fissures*. MIM. : Série Matériaux et métallurgie. Hermes Science Publications, 2009.
- [Baum 1970] Leonard E. Baum, Ted Petrie, George Soules et Norman Weiss. *A Maximization Technique Occurring in the Statistical Analysis of Probabi-*

- listic Functions of Markov Chains*. The Annals of Mathematical Statistics, vol. 41, no. 1, pages pp. 164–171, 1970.
- [Baussaron 2010] Julien Baussaron, Barreau Mihaela, Gerville reache Léo, Guerin Fabrice et Schimmerling Paul. *Estimation de la fiabilité à partir de mesures de dégradation : quelles méthodes utiliser?* La, 17eme Congrès de Maîtrise des Risques et de Sûreté de Fonctionnement, octobre 2010.
- [Bellman 1971] R.E. Bellman et S.E. Dreyfus. *Applied dynamic programming*. Princeton University Press, 1971.
- [Ben-Daya 2000] Mohamed Ben-Daya, Salih O. Duffaa et Abdul Raouf. *Maintenance, modeling, and optimization*. Kluwer Academic Publisher, Massachusetts, 2000.
- [Berthiau 2001] G. Berthiau et P. Siarry. *État de l'art des méthodes d'optimisation globales*. RAIRO-Operations Research, vol. tom35, no. 3, pages 329–365, 2001.
- [Bishop 2002] Christopher M. Bishop, David Spiegelhalter et John Winn. *VIBES : A Variational Inference Engine for Bayesian Networks*. In *NIPS 2002*, 2002.
- [Blum 2005] Christian Blum. *Ant colony optimization : Introduction and recent trends*. Physics of Life Reviews, vol. 2, no. 4, pages 353–373, Décembre 2005.
- [Boudali 2005] H. Boudali et J. B. Dugan. *A discrete-time Bayesian network reliability modeling and analysis framework*. Reliability Engineering & System Safety, vol. 87, no. 3, pages 337–349, March 2005.
- [Bouillaut 2009] L. Bouillaut, R. Donat, A. Neji et P. Aknin. *Estimation of Multi-component Systems reliability : comparison of two Graphical Model Approaches*. In *proceedings of the 13th IFAC Symposium on Information Control Problems in Manufacturing*, June 2009.
- [Bouillaut 2011] L. Bouillaut, O. François, P. Aknin, R. Donat, S. Bondeux et S. Dubois. *VirMaLab - atelier virtuel de maintenance : un outil d'aide à la décision pour l'optimisation des politiques de maintenance*. Recherche, transports, sécurité, no. 107, pages 241–257, 2011.
- [Bouvard 2011] K. Bouvard, S. Artus, C. Bérenguer et V. Cocquempot. *Condition-based dynamic maintenance operations planning & grouping. Application to commercial heavy vehicles*. Reliability Engineering & System Safety, vol. 96, no. 6, pages 601 – 610, 2011.
- [Boyen 1999] Xavier Boyen et Daphne Koller. *Approximate Learning of Dynamic Models*. In Michael S. Kearns, Sara A. Solla et David A. Kohn, éditeurs,

- Advances in Neural Information Processing Systems 11 : Proceedings of the 1998 Conference—NIPS 1998, pages 396–402. Cambridge : MIT Press, 1999. Available at <http://www.cs.stanford.edu/~xb/nips98/>.
- [Bremaud 2001] P. Bremaud. Markov Chains : Gibbs Fields, Monte Carlo Simulation, and Queues. Springer-Verlag New York Inc., corrected édition, February 2001.
- [Brown 1994] J.D. Brown. Proceedings of the cornelius lanczos international centenary conference. Proceedings in Applied Mathematics Series. Society for Industrial & Applied, 1994.
- [Campbell 2001] John D. Campbell et Andrew K. S. Jardine. Maintenance Excellence : Optimizing Equipment Life-Cycle Decisions (Dekker Mechanical Engineering). CRC Press, 1 édition, Février 2001.
- [Castanier 2001] B. Castanier. Modélisation stochastique et optimisation de la maintenance conditionnelle des systèmes à dégradation graduelle. 2001.
- [Castanier 2003] B. Castanier, C. Berenguer et A. Grall. *A sequential condition-based repair/replacement policy with non-periodic inspections for a system subject to continuous wear*. Applied stochastic models in business and industry, vol. 19, no. 4, pages 327–347, 2003.
- [Castanier 2005] B. Castanier, A. Grall et C. Bérenguer. *A condition-based maintenance policy with non-periodic inspections for a two-unit series system*. Reliability Engineering & System Safety, vol. 87, no. 1, pages 109–120, 2005.
- [Castro 2009] I.T. Castro. *A model of imperfect preventive maintenance with dependent failure modes*. European Journal of Operational Research, vol. 196, no. 1, pages 217–224, 2009.
- [Celeux 2006] G. Celeux, F. Corset, A. Lannoy et B. Ricard. *Designing a Bayesian network for preventive maintenance from expert opinions in a rapid and reliable way*. Reliability Engineering & System Safety, vol. 91, no. 7, pages 849–856, July 2006.
- [Cerny 1985] V. Cerny. *Thermodynamical approach to the traveling salesman problem : An efficient simulation algorithm*. Journal of Optimization Theory and Applications, vol. 45, no. 1, pages 41–51, January 1985.
- [Chan 2001] Weng-Tat Chan et Hao Hu. *An application of genetic algorithms to precast production scheduling*. Computers & Structures, vol. 79, no. 17, pages 1605–1616, 2001.
- [Chan 2006] G.K. Chan et S. Asgarpoor. *Optimum maintenance policy with Markov processes*. Electric Power Systems Research, vol. 76, no. 6–7, pages 452–456, 2006.

- [Chang 2007] Qing Chang, Jun Ni, Pulak Bandyopadhyay, Stephan Biller et Guoxian Xiao. *Maintenance Opportunity Planning System*. Journal of Manufacturing Science and Engineering-transactions of The Asme, vol. 129, 2007.
- [Chapot 1999] Jorge Luiz C Chapot, Fernando Carvalho Da Silva et Roberto Schirru. *A new approach to the use of genetic algorithms to solve the pressurized water reactor's fuel management optimization problem*. Annals of Nuclear Energy, vol. 26, no. 7, pages 641–655, 1999.
- [Chen 2005] Dongyan Chen et Kishor S. Trivedi. *Optimization for condition-based maintenance with semi-Markov decision process*. Reliability Engineering & System Safety, vol. 90, no. 1, pages 25–29, 2005.
- [Chipperfield 1994] A. Chipperfield, P. Fleming, H. Pohlheim et C. Fonseca. *Genetic Algorithm Toolbox : User Guide*. Rapport technique, Department of Automatic Control and Systems Engineering of The University of Sheffield, UK, 1994.
- [Coello 1998] Carlos A. Coello Coello et Laboratorio Nacional. *An Updated Survey of GA-Based Multiobjective Optimization Techniques*. ACM Computing Surveys, vol. 32, pages 109–143, 1998.
- [Coello 2005] Carlos A. Coello Coello, Gregorio Toscano Pulido et Efrén Mezura Montes. *Current and Future Research Trends in Evolutionary Multiobjective Optimization*, 2005.
- [Coello 2006] Carlos A. Coello Coello, Gary B. Lamont et David A. Van Veldhuizen. *Evolutionary algorithms for solving multi-objective problems (genetic and evolutionary computation)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [Coit 1996a] David W. Coit, Ieee S. Member, Alice E. Smith et Ieee Member. *Reliability Optimization of Series-Parallel Systems Using a Genetic Algorithm*. IEEE Transactions on Reliability, vol. 45, pages 254–260, 1996.
- [Coit 1996b] David W. Coit et Alice E. Smith. *Penalty Guided Genetic Search For Reliability Design Optimization*. Computers and Industrial Engineering, vol. 30, pages 895–904, 1996.
- [Collette 2004] Y. Collette et P. Siarry. *Multiobjective optimization : principles and case studies*. Springer-Verlag, 2004.
- [Coloni 1991] Alberto Coloni, Marco Dorigo et Vittorio Maniezzo. *Distributed Optimization by Ant Colonies*. In European Conference on Artificial Life, pages 134–142, 1991.

- [Cooper 1990] Gregory F. Cooper. *The Computational Complexity of Probabilistic Inference Using Bayesian Belief Networks*. Artificial Intelligence, vol. 42, no. 2-3, pages 393–405, 1990.
- [Corriou 2010] Jean-Pierre Corriou. *Méthodes numériques et optimisation - Théorie et pratique pour l'ingénieur*. Lavoisier, Tec. & Doc., 2010.
- [Corset 2003] Franck Corset. *Aide à l'optimisation de maintenance à partir de réseaux bayésiens et fiabilité dans un contexte doublement censuré*. PhD thesis, Université Joseph Fourier, 2003.
- [Courtois 2006] P. Courtois et P. Delsarte. *On the optimal scheduling of periodic tests and maintenance for reliable redundant components*. Reliability Engineering & System Safety, vol. 91, no. 1, pages 66–72, January 2006.
- [Crowder 2007] Martin Crowder et Jerald Lawless. *On a scheme for predictive maintenance*. European Journal of Operational Research, vol. 176, no. 3, pages 1713–1722, Février 2007.
- [D'Ambrosio 1993] Bruce D'Ambrosio. *Incremental Probabilistic Inference*. In Proceedings of the Ninth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-93), pages 301–308, San Francisco, CA, 1993. Morgan Kaufmann.
- [Dantzig 1987] G.B. Dantzig et Stanford University. *Systems Optimization Laboratory. Origins of the simplex method*. Technical report (Stanford University. Systems Optimization Laboratory). Defense Technical Information Center, 1987.
- [Darwiche 2001] Adnan Darwiche. *Constant-space reasoning in dynamic Bayesian networks*. International Journal of Approximate Reasoning, vol. 26, no. 3, pages 161 – 178, 2001.
- [Das 1997] I. Das et J. E. Dennis. *A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems*. Structural and Multidisciplinary Optimization, vol. 14, no. 1, pages 63–69, Août 1997.
- [De Jong 2007] K. De Jong. *Parameter Setting in EAs : a 30 Year Perspective*. In F.G. Lobo, C.F. Lima et Z. Michalewicz, éditeurs, *Parameter Setting in Evolutionary Algorithms*, pages 1–18. Springer, 2007.
- [Dean 1989] Thomas Dean et Keiji Kanazawa. *A model for reasoning about persistence and causation*. Computational Intelligence, vol. 5, no. 2, pages 142–150, 1989.
- [Deb 2000] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap et T Meyarivan. *A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective*

- Optimization : NSGA-II*. In Marc Schoenauer, Kalyanmoy Deb, Günther Rudolph, Xin Yao, Evelyne Lutton, Juan Merelo et Hans-Paul Schwefel, éditeurs, *Parallel Problem Solving from Nature PPSN VI*, volume 1917 of *Lecture Notes in Computer Science*, pages 849–858. Springer Berlin / Heidelberg, 2000.
- [Deb 2003] Kalyanmoy Deb, Manikant Mohan et Shikhar Mishra. *Towards a Quick Computation of Well-Spread Pareto-Optimal Solutions*. In CarlosM. Fonseca, PeterJ. Fleming, Eckart Zitzler, Lothar Thiele et Kalyanmoy Deb, éditeurs, *Evolutionary Multi-Criterion Optimization*, volume 2632 of *Lecture Notes in Computer Science*, pages 222–236. Springer Berlin Heidelberg, 2003.
- [Dechter 1999] Rina Dechter. *Bucket elimination : A unifying framework for reasoning*. *Artificial Intelligence*, vol. 113, no. 1-2, pages 41 – 85, 1999.
- [Dekker 1996] Rommert Dekker. *Applications of maintenance optimization models : a review and analysis*. *Reliability Engineering & System Safety*, vol. 51, no. 3, pages 229–240, Mars 1996.
- [Deloux 2008] E. Deloux. *Politiques de maintenance conditionnelle pour un système à dégradation continue soumis à un environnement stressant*. 2008.
- [Dieulle 2003] L Dieulle, C Bérenguer, A Grall et M Roussignol. *Sequential condition-based maintenance scheduling for a deteriorating system*. *European Journal of Operational Research*, vol. 150, no. 2, pages 451–461, 2003.
- [Do Van 2012] Phuc Do Van, Hai Canh Vu, Anne Barros et Christophe Berenguer. *Grouping maintenance strategy with availability constraint under limited repairmen*. In 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, SAFEPROCESS 2012, page Paper ThFtc11C.4, Mexico city, Mexique, Août 2012. SYMPA, ISET SYMPA, ISET.
- [Doksum 1992] Kjell A. Doksum, Arnljot Høyland et Arnljot Hoyland. *Models for Variable-Stress Accelerated Life Testing Experiments Based on Wiener Processes and the Inverse Gaussian Distribution*. *Technometrics*, vol. 34, no. 1, pages 74–82, Février 1992.
- [Doksum 1995] Kjell A. Doksum et Sharon-Lise T. Normand. *Gaussian models for degradation processes-part I : Methods for the analysis of biomarker data*. *Lifetime Data Analysis*, vol. 1, no. 2, pages 131–144, Juin 1995.
- [Donat 2009] Roland Donat. *Modélisation de la fiabilité et de la maintenance par modèles graphiques probabilistes Application à la prévention des ruptures de rails*. PhD thesis, INSA Roune, 2009.



- [Donat 2010] Roland Donat, Philippe Leray, Laurent Bouillaut et Patrice Aknin. *A dynamic Bayesian network to represent discrete duration models*. Neurocomput., vol. 73, no. 4-6, pages 570–577, Janvier 2010.
- [Draper 1995] Denise Draper et Steve Hanks. *Localized Partial Evaluation of Belief Networks*. pages 170–177. Morgan Kaufmann, 1995.
- [Dréo 2005] J. Dréo, A. Pétrowski, P. Siarry et É. Taillard. Métaheuristique pour l'optimisation difficile, volume Eyrolles. Eyrolles, Paris, 2ème tirage édition, 2005. ouvrage coordonné par Patrick Siarry.
- [Durand 1994] Nicolas Durand, Jean-Marc Alliot et Joseph Noailles. *Algorithmes Génétiques : un croisement pour les problèmes partiellement séparables*. In Evolution Artificielle 94 (Toulouse), 1994.
- [Engrand 1997] P. Engrand. *A multi-objective optimization approach based on simulated annealing and its application to nuclear fuel management*. In Proceedings of the 5<sup>th</sup> International Conference on Nuclear Engineering, pages 416–423. American Society Of Mechanical Engineers, 1997.
- [Eshelman 1992] Larry J. Eshelman et J. David Schaffer. In L. Darrell Whitley, éditeur, FOGA, pages 187–202. Morgan Kaufmann, 1992.
- [Feldman 1977] R. M. Feldman. *Optimal Replacement with Semi-Markov Shock Models Using Discounted Costs*. Mathematics of Operations Research, vol. 2, pages 78–90, 1977.
- [Fogel 1966] L. J. Fogel, A. J. Owens et M. J. Walsh. Artificial intelligence through simulated evolution. John Wiley, New York, USA, 1966.
- [Folks 1978] J. L. Folks et R. S. Chhikara. *The Inverse Gaussian Distribution and Its Statistical Application—A Review*. Journal of the Royal Statistical Society. Series B (Methodological), vol. 40, no. 3, pages pp. 263–289, 1978.
- [Gil 1995] Markov chain monte carlo in practice (chapman & Hall/CRC interdisciplinary statistics). Chapman and Hall/CRC, 1 édition, Décembre 1995.
- [Glover 1986] F. Glover. *Future paths for integer programming and links to artificial intelligence*. Comput. Oper. Res., vol. 13, no. 5, pages 533–549, May 1986.
- [Goldberg 1989] D.E Goldberg. Genetic algorithms in search, optimization and machine learning. Addison-Wesley, 1989.
- [Gordon 1993] N.J. Gordon, D.J. Salmond et A.F.M. Smith. *Novel approach to nonlinear/non-Gaussian Bayesian state estimation*. IEEE Proceedings F, Radar and Signal Processing, vol. 140, no. 2, pages 107–113, 1993.
- [Gosselin 2009] L. Gosselin, M Tye-Gingras et F Mathieu-Potvin. *Review of utilization of genetic algorithms in heat transfer problems*. International

- Journal of Heat and Mass Transfer, vol. 52, no. 9-10, pages 2169–2188, April 2009.
- [Grall 2002] A. Grall, C. Bérenguer et L. Dieulle. *A condition-based maintenance policy for stochastically deteriorating systems*. Reliability Engineering & System Safety, vol. 76, no. 2, pages 167–180, May 2002.
- [Grall 2006] A. Grall, L. Dieulle, C. Bérenguer et M. Roussignol. *Asymptotic failure rate of a continuously monitored system*. Reliability Engineering & System Safety, vol. 91, no. 2, pages 126 – 130, 2006. Selected Papers Presented at QUALITA 2003.
- [Grefenstette 1986] J. J. Grefenstette. *Optimization of Control Parameters for Genetic Algorithms*. Systems, Man and Cybernetics, IEEE Transactions on, vol. 16, no. 1, pages 122–128, Janvier 1986.
- [Grigoriev 2006] A. Grigoriev, J. Vandeklundert et F. Spieksma. *Modeling and solving the periodic maintenance problem*. European Journal of Operational Research, vol. 172, no. 3, pages 783–797, August 2006.
- [Guéret 2000] C. Guéret, C. Prins et M. Sevaux. *Programmation linéaire : 65 problèmes d’optimisation modélisés et résolus avec Visual Xpress*. Algorithmes (Paris). Eyrolles, 2000.
- [Hajji 2003] O. Hajji. *Contribution au développement de méthodes d’optimisation stochastiques : application à la conception des dispositifs électrotechniques*. 2003.
- [Hansen 2001] Nikolaus Hansen et Andreas Ostermeier. *Completely Derandomized Self-Adaptation in Evolution Strategies*. Evol. Comput., vol. 9, no. 2, pages 159–195, Juin 2001.
- [Holland 1992] J. H. Holland. *Adaptation in natural and artificial systems*. MIT Press, Cambridge, MA, USA, 1992.
- [Howard 1984] Ronald A. Howard et James E. Matheson. *Influence Diagrams*. Decision Analysis, September 2005, vol. 2, pages 127–143, 1984.
- [Howard 2007] R.A. Howard. *Dynamic probabilistic systems, volume ii : Semi-markov and decision processes*. Dover Books on Mathematics Series. Dover Publications, 2007.
- [Igel 2007] Christian Igel, Nikolaus Hansen et Stefan Roth. *Covariance Matrix Adaptation for Multi-objective Optimization*. Evol. Comput., vol. 15, no. 1, pages 1–28, Mars 2007.
- [Isard 1996] Michael Isard et Andrew Blake. *Contour Tracking By Stochastic Propagation of Conditional Density*. pages 343–356, 1996.

- [Jin 2001] Yaochu Jin, Markus Olhofer et Bernhard Sendhoff. *Dynamic Weighted Aggregation for Evolutionary Multi-Objective Optimization : Why Does It Work and How?* In Proceedings of the Genetic and Evolutionary Computation Conference GECCO, pages 1042–1049. Morgan Kaufmann, 2001.
- [Jong 1992] Kenneth A. De Jong, Kenneth A. De, Jong William et William M. Spears. *A Formal Analysis of the Role of Multi-Point Crossover in Genetic Algorithms*, 1992.
- [Jordan 1999] Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola et Lawrence K. Saul. *An Introduction to Variational Methods for Graphical Models*. Mach. Learn., vol. 37, no. 2, pages 183–233, Novembre 1999.
- [Kalman 1960] R. E. Kalman. *A New Approach to Linear Filtering and Prediction Problems*. Transactions of the ASME - Journal of Basic Engineering, no. 82 (Series D), pages 35–45, 1960.
- [Kang 1999] C. W. Kang et M. W. Golay. *A Bayesian belief network-based advisory system for operational availability focused diagnosis of complex nuclear power systems*. Expert Systems with Applications, vol. 17, no. 1, pages 21–32, July 1999.
- [Kesen 2010] Saadettin Erhan Kesen, Sanchoy K. Das et Zülal Güngör. *A genetic algorithm based heuristic for scheduling of virtual manufacturing cells (VMCs)*. Computers & Operations Research, vol. 37, no. 6, pages 1148–1156, 2010.
- [Kim 2009] Michael Jong Kim et Viliam Makis. *Optimal maintenance policy for a multi-state deteriorating system with two types of failures under general repair*. Computers & Industrial Engineering, vol. 57, no. 1, pages 298 – 303, 2009.
- [Kirkpatrick 1983] S. Kirkpatrick, C. D. Gelatt et M. P. Vecchi. *Optimization by Simulated Annealing*. Science, vol. 220, no. 4598, pages 671–680, May 1983.
- [Kjoerulff 1994] Uffe Kjoerulff. *Reduction of Computational Complexity in Bayesian Networks through Removal of Weak Dependences*. In IN PROC. TENTH CONF. ON UNCERTAINTY IN ARTIFICIAL INTELLIGENCE, pages 374–382. Morgan Kaufmann, 1994.
- [Kolowrocki 2011] K. Kolowrocki et J. Soszynska-Budny. *Reliability and Safety of Complex Technical Systems and Processes : Modeling-identification-prediction-optimization*. Springer Series in Reliability Engineering. Springer, 2011.

- [Kong 1997] Myung B. Kong et Kyung S. Park. *Optimal replacement of an item subject to cumulative damage under periodic inspections*. Microelectronics Reliability, vol. 37, no. 3, pages 467–472, 1997.
- [Kumar 2012] Naveen Kumar, Jin-Hwan Borm et Ajay Kumar. *Reliability analysis of waste clean-up manipulator using genetic algorithms and fuzzy methodology*. Computers & Operations Research, vol. 39, no. 2, pages 310–319, 2012.
- [Lacomme 2003] P. Lacomme, C. Prins et M. Sevaux. Algorithmes de graphes. Algorithmes (Paris). Eyrolles, 2003.
- [Laggoune 2009] Radouane Laggoune, Alaa Chateauneuf et Djamil Aissani. *Opportunistic policy for optimal preventive maintenance of a multi-component system in continuous operating units*. Computers & Chemical Engineering, vol. 33, no. 9, pages 1499 – 1510, 2009.
- [Lam 2003] Yeh Lam et Yuan L. Zhang. *A geometric-process maintenance model for a deteriorating system under a random environment*. IEEE Transactions on Reliability, vol. 52, no. 1, pages 83–89, Mars 2003.
- [Langseth 2007] H. Langseth et L. Portinale. *Bayesian networks in reliability*. Reliability Engineering & System Safety, vol. 92, no. 1, pages 92–108, January 2007.
- [Lapa 2000] C. Lapa. *Maximization of a nuclear system availability through maintenance scheduling optimization using a genetic algorithm*. Nuclear Engineering and Design, vol. 196, no. 2, pages 219–231, March 2000.
- [Laumanns 2002] Marco Laumanns, Lothar Thiele, Kalyanmoy Deb et Eckart Zitzler. *Combining convergence and diversity in evolutionary multiobjective optimization*. Evol. Comput., vol. 10, no. 3, pages 263–282, Septembre 2002.
- [Lauritzen 1988] S. L. Lauritzen et D. J. Spiegelhalter. *Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems*. Journal of the Royal Statistical Society. Series B (Methodological), vol. 50, no. 2, 1988.
- [Lauritzen 1996] Steffen L. Lauritzen. Graphical models. Oxford University Press, 1996.
- [Lawrence 2000] Neil D. Lawrence. *Variational Inference in Probabilistic Models*. PhD thesis, Cambridge University, 2000.
- [Levitin 1998] G. Levitin, A. Lisnianski, H. Ben-Haim et D. Elmakis. *Redundancy optimization for series-parallel multi-state systems*. Reliability, IEEE Transactions on, vol. 47, no. 2, pages 165–172, jun 1998.

- [Li 1994] Zhaoyu Li et Bruce D'Ambrosio. *Efficient inference in Bayes networks as a combinatorial optimization problem*. International Journal of Approximate Reasoning, vol. 11, no. 1, pages 55 – 81, 1994.
- [Liao 2006] Haitao Liao et Elsayed A. Elsayed. *Reliability inference for field conditions from accelerated degradation testing*. Naval Research Logistics (NRL), vol. 53, no. 6, pages 576–587, 2006.
- [Lopez 2001] P. Lopez et F. Roubellat. Ordonnancement de la production. 2001.
- [Lundy 1986] M. Lundy et A. Mees. *Convergence of an annealing algorithm*. vol. 34, no. 1, pages 111–124, 1986.
- [Luo 2008] Biao Luo, Jinhua Zheng, Jiongliang Xie et Jun Wu. *Dynamic Crowding Distance? A New Diversity Maintenance Strategy for MOEAs*. In Natural Computation, 2008. ICNC '08. Fourth International Conference on, volume 1, pages 580 –585, oct. 2008.
- [Marseguerra 2002] M. Marseguerra, E. Zio et L. Podofillini. *Condition-based maintenance optimization by means of genetic algorithms and Monte Carlo simulation*. Reliability Engineering & System Safety, vol. 77, no. 2, pages 151–165, August 2002.
- [Marseguerra 2004] Marzio Marseguerra, Enrico Zio et Raffaele Canetta. *Using genetic algorithms for calibrating simplified models of nuclear reactor dynamics*. Annals of Nuclear Energy, vol. 31, no. 11, pages 1219–1250, 2004.
- [Martínez-Iranzo 2009] Miguel Martínez-Iranzo, Juan M. Herrero, Javier Sanchis, Xavier Blasco et Sergio García-Nieto. *Applied Pareto multi-objective optimization by stochastic solvers*. Eng. Appl. Artif. Intell., vol. 22, no. 3, pages 455–465, Avril 2009.
- [Miettinen 1999] K. Miettinen. Nonlinear multiobjective optimization. International Series in Operations Research & Management Science, 12. Kluwer Academic Pub, 1999.
- [Mitchell 1996] M. Mitchell. An introduction to genetic algorithms. MIT Press, Cambridge, MA, USA, 1996.
- [Murphy 1999] K. P. Murphy, Y. Weiss et M. I. Jordan. *Loopy Belief Propagation for Approximate Inference : An Empirical Study*. In Proceedings of the 15th on Uncertainty in Artificial Intelligence, pages 467–475, 1999.
- [Murphy 2002] P. Murphy. *Dynamic Bayesian Networks : Representation, Inference and Learning*. PhD thesis, University of California, Berkeley, 2002.
- [Nachlas 2005] J.A. Nachlas. Reliability engineering : Probabilistic models and maintenance methods. Mechanical Engineering. Taylor & Francis Group, 2005.

- [Nakagawa 1989] T. Nakagawa. *A summary of replacement models with changing failure distributions*. 1989.
- [Nakagawa 2005] Toshio Nakagawa. *Maintenance theory of reliability*. Springer-Verlag, London, 2005.
- [Nakagawa 2010] T. Nakagawa. *Advanced Reliability Models and Maintenance Policies*. Springer Series in Reliability Engineering. Springer, 2010.
- [Nelson 1990] Wayne B. Nelson. *Accelerated Testing : Statistical Models, Test Plans, and Data Analysis* (Wiley Series in Probability and Statistics). Wiley-Interscience, 1 édition, 1990.
- [Nikulin 2006] Mikhail Nikulin. *Statistique des essais accélérés* (french edition). Hermes science publications, 2006.
- [Noortwijk 2009] J. M. Van Noortwijk. *A survey of the application of gamma processes in maintenance*. *Reliability Engineering & System Safety*, vol. 94, no. 1, pages 2–21, Janvier 2009.
- [Okabe 2003] T. Okabe, Y. Jin et B. Sendhoff. *A critical survey of performance indices for multi-objective optimisation*. In *Evolutionary Computation, 2003. CEC '03. The 2003 Congress on*, volume 2, pages 878–885 Vol.2, 2003.
- [Osaki 2002] Shunji Osaki. *Stochastic models in reliability and maintenance*. Springer, Berlin, 2002.
- [Padgett 2004] W. J. Padgett et Meredith A. Tomlinson. *Inference from Accelerated Degradation and Failure Data Based on Gaussian Process Models*. *Lifetime Data Analysis*, vol. 10, no. 2, pages 191–206, Juin 2004.
- [Park 1988] K. S. Park. *Optimal continuous-wear limit replacement under periodic inspections*. *IEEE Transactions on Reliability*, vol. 37, no. 1, pages 97–102, Avril 1988.
- [Pearl 1985] Judea Pearl. *Bayesian networks : A model of self-activated memory for evidential reasoning*. In *Proceedings of the 7th Conference of the Cognitive Science Society, University of California, Irvine*, pages 329–334, Août 1985.
- [Pearl 1986] Judea Pearl. *Fusion, Propagation, and Structuring in Belief Networks*, 1986.
- [Pereira 2003] C. Pereira. *Coarse-grained parallel genetic algorithm applied to a nuclear reactor core design optimization problem*. *Annals of Nuclear Energy*, vol. 30, no. 5, pages 555–565, March 2003.
- [Pereira 2008] Cláudio M.N.A. Pereira et Wagner F. Sacco. *A parallel genetic algorithm with niching technique applied to a nuclear reactor core design optimization problem*. *Progress in Nuclear Energy*, vol. 50, no. 7, pages 740–746, 2008.

- [Pham 2006] H. Pham. Reliability modeling, analysis and optimization. Series on quality, reliability & engineering statistics. World Scientific, 2006.
- [Pierreval 1997] H. Pierreval et L. Tautou. *Using evolutionary algorithms and simulation for the optimization of manufacturing systems*. IIE Transactions, vol. 29, no. 3, pages 181–189, Mars 1997.
- [Qian 2003] C. Qian, S. Nakamura et T. Nakagawa. *Replacement and minimal repair policies for a cumulative damage model with maintenance*. Computers & Mathematics with Applications, vol. 46, no. 7, pages 1111 – 1118, 2003. <ce :title>Applied Stochastic System Modelling</ce :title>.
- [Rabiner 1986] L. R. Rabiner et B. H. Juang. *An introduction to hidden Markov models*. IEEE ASSp Magazine, 1986.
- [Ramesh 2012] S. Ramesh, S. Kannan et S. Baskar. *Application of modified NSGA-II algorithm to multi-objective reactive power planning*. Applied Soft Computing, vol. 12, no. 2, pages 741 – 753, 2012.
- [Randelman 1986] R.E. Randelman et GaryS. Grest. *N-city traveling salesman problem : Optimization by simulated annealings*. Journal of Statistical Physics, vol. 45, pages 885–890, 1986.
- [Reeves 2002] Colin R. Reeves et Jonanthan E.Rowe. Genetic algorithms principles and perspectives A guide to GA theory. Kluwer Academic Publisher, 2002.
- [Rémi 2010] Rhodes Rémi. *Processus de Lévy et calcul stochastique*, novembre 2010.
- [Revuz 2004] Daniel Revuz et Marc Yor. Continuous martingales and brownian motion (grundlehren der mathematischen wissenschaften). Springer, 3rd édition, Décembre 2004.
- [Rosenberg 1967] R.S. Rosenberg. *Simulation of genetic populations with biochemical properties*. PhD thesis, Univ. Michigan, 1967.
- [Ross 1995] Sheldon M. Ross. Stochastic Processes. Wiley, 2 édition, Janvier 1995.
- [Sacco 2002] Wagner F Sacco, Cláudio M.N.A Pereira, Pius P.M Soares et Roberto Schirru. *Genetic algorithms applied to turbine extraction optimization of a pressurized-water reactor*. Applied Energy, vol. 73, no. 3–4, pages 217–222, 2002.
- [Sakarovitch 1984] M. Sakarovitch. Optimisation combinatoire : . Graphes et programmation linéaire. Collection Enseignement des sciences. Hermann, 1984.

- [Santos 2010] Maristela Oliveira Santos, Sadao Massago et Bernardo Almada-Lobo. *Infeasibility handling in genetic algorithm using nested domains for production planning*. Computers & Operations Research, vol. 37, no. 6, pages 1113–1122, 2010.
- [Schaffer 1985] J. David Schaffer. *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*. In Proceedings of the 1st International Conference on Genetic Algorithms, pages 93–100, Hillsdale, NJ, USA, 1985. L. Erlbaum Associates Inc.
- [Sheu 1997] Shey-Huei Sheu et Jhy-Ping Jhang. *A generalized group maintenance policy*. European Journal of Operational Research, vol. 96, no. 2, pages 232 – 247, 1997.
- [Smith 1989] J. Q. Smith. *Influence Diagrams for Statistical Modelling*. The Annals of Statistics, vol. 17, no. 2, pages 654–672, 1989.
- [Soro 2010] Isaac W. Soro, Mustapha Noureldath et Daoud AËt-Kadi. *Performance evaluation of multi-state degraded systems with minimal repairs and imperfect preventive maintenance*. Reliability Engineering & System Safety, vol. 95, no. 2, pages 65–69, 2010.
- [Spears 1991] William M. Spears et Kenneth. *An Analysis of Multi-Point Crossover*. In G. J. E. Rawlins, editeur, Foundations of Genetic Algorithms, pages 301–315, San Mateo, CA, 1991. Morgan Kaufmann.
- [Srinivas 1994] N. Srinivas et Kalyanmoy Deb. *Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms*. Evolutionary Computation, vol. 2, pages 221–248, 1994.
- [Terrier 2002] N. Terrier. *La maintenance*. ISTIA - Département Qualité, 2002.
- [Tsai 2001] Y Tsai. *Optimizing preventive maintenance for mechanical components using genetic algorithms*. Reliability Engineering & System Safety, vol. 74, no. 1, pages 89–97, October 2001.
- [Ulungu 1999] E.L. Ulungu, J. Teghem, P.H. Fortemps et D. Tuytens. *MOSA method : a tool for solving multiobjective combinatorial optimization problems*. Journal of Multi-Criteria Decision Analysis, vol. 8, no. 4, pages 221–236, 1999.
- [Villmeur 1988] A. Villmeur. Sûreté de fonctionnement des systèmes industriels, Fiabilité - Facteurs Humains - Informatisation. Paris, édition eyrolles édition, 1988.
- [Wang 2002] Hongzhou Wang. *A survey of maintenance policies of deteriorating systems*. European Journal of Operational Research, vol. 139, no. 3, pages 469 – 489, 2002.



- [Whitmore 1998] G. A. Whitmore, M. J. Crowder et J. F. Lawless. *Failure inference from a marker process based on a bivariate Wiener model*. Lifetime data analysis, vol. 4, no. 3, pages 229–251, 1998.
- [Wildeman 1996] R.E. Wildeman. The art of grouping maintenance : De kunst van het onderhoud groeperen. Tinbergen Institute research series : Tinbergen Instituut. Thesis, 1996.
- [Yang 1999] Joon-Eon Yang, Mee-Jung Hwang, Tae-Yong Sung et Youngho Jin. *Application of genetic algorithm for reliability allocation in nuclear power plants*. Reliability Engineering & System Safety, vol. 65, no. 3, pages 229–238, 1999.
- [Yeh 2009] Ruey H. Yeh, Kow-Chin Kao et Wen L. Chang. *Optimal preventive maintenance policy for leased equipment using failure rate reduction*. Computers & Industrial Engineering, vol. 57, no. 1, pages 304–309, August 2009.
- [Zitzler 1999a] E Zitzler et L Thiele. *Multiobjective evolutionary algorithms : a comparative case study and the strength Pareto approach*. IEEE Transactions on Evolutionary Computation, vol. 3, no. 4, pages 257–271, 1999.
- [Zitzler 1999b] Eckart Zitzler. *Evolutionary Algorithms for Multiobjective Optimization : Methods and Applications*, 1999.
- [Zitzler 2000] Eckart Zitzler, Kalyanmoy Deb et Lothar Thiele. *Comparison of Multiobjective Evolutionary Algorithms : Empirical Results*. Evolutionary Computation, vol. 8, pages 173–195, 2000.
- [Zitzler 2002] Eckart Zitzler, Marco Laumanns et Lothar Thiele. *SPEA2 : Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization*. In Evolutionary Methods for Design, Optimisation, and Control, pages 95–100. CIMNE, Barcelona, Spain, 2002.
- [Zweig 1996] G. Zweig. A forward-backward algorithm for inference in bayesian networks and an empirical comparison with HMMs. Master's thesis, UC Berkeley CS Dept, 1996.
- [Zwingelstein 1996] G. Zwingelstein. La maintenance basée sur la fiabilité : Guide pratique d'application de la rcm. Collection Diagnostic et maintenance. Hermès, 1996.